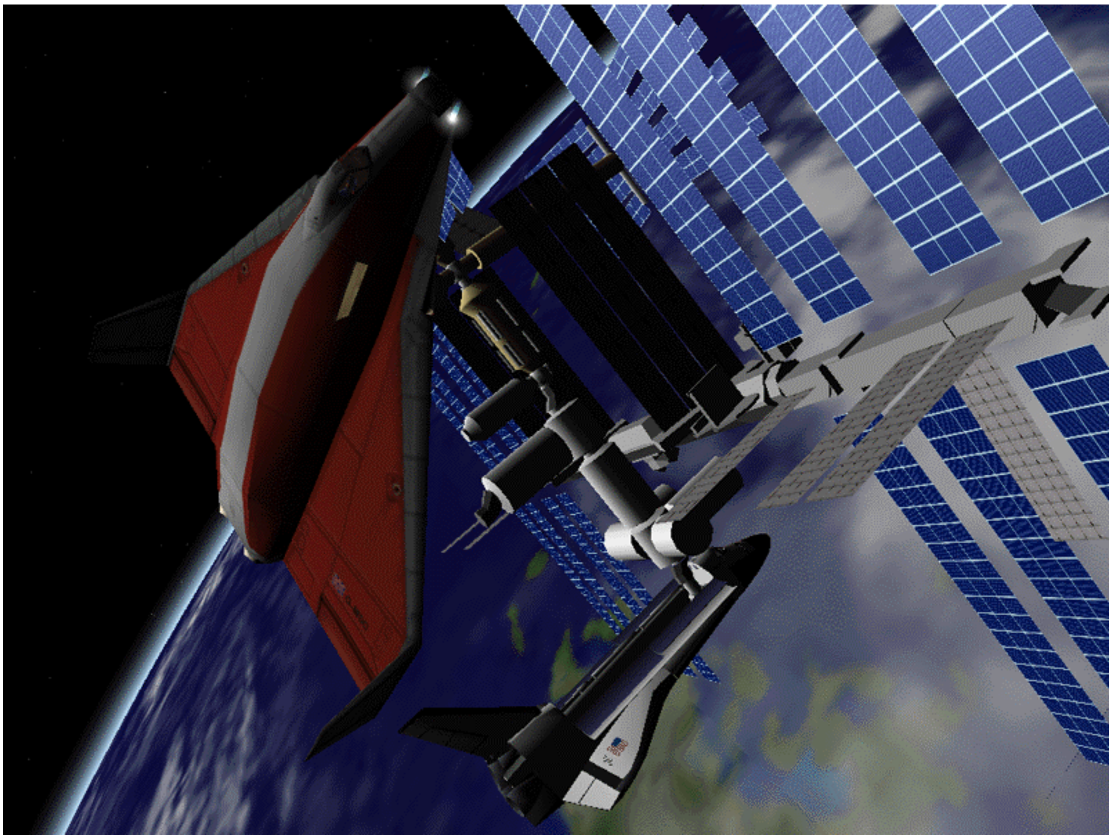
ORBITER User Manual

©2000-2002 Martin Schweiger
martins@medphys.ucl.ac.uk
www.medphys.ucl.ac.uk/~martins/orbit/orbit.html
19 April 2002



Disclaimer

The ORBITER software is provided as is without any warranty of any kind. To the maximum extent permitted by applicable law, Martin Schweiger further disclaims all warranties, including without limitation any implied or stated warranties of merchantability, fitness for a particular purpose, and noninfringement. The entire risk arising out of the use or performance of this product and documentation remains with recipient. To the maximum extent permitted by applicable law, in no event shall program ORBITER or its suppliers be liable for any consequential, incidental, direct, indirect, special, punitive, recursive, or other damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, personal injury, disruption of family life, or other pecuniary loss) arising out of this agreement or the use of or inability to use the product, even if program ORBITER has been advised of the possibility of such damages. Because some states/jurisdictions do not allow the exclusion or limitation of liability of consequential or incidental damages, the above limitation may not apply to the recipient.

Contents

1. Introduction

ORBITER is a space flight simulator which uses realistic physical models for planetary motion, spacecraft dynamics and atmospheric effects. Use sophisticated instruments and navigation computers to travel between spaceports on planet surfaces and orbital stations throughout the solar system. ORBITER is highly configurable and allows the modification of existing objects as well as the addition of new spacecraft, orbital stations, surface bases or the construction of a new planetary system from scratch.

ORBITER is not an arcade game. To make the most of it, a basic understanding of physics is required.

2. Support

Suggestions, corrections, bug reports (and praise) for the ORBITER software or this documentation are always welcome. The preferred way to post your comments, in particular if they may be of interest to other users, is via the ORBITER mailing list or the ORBITER web forum, available via links from the official Orbiter site www.orbitersim.com. I will also try to answer mail sent to directly to me, but I haven't really got the time or resources to officially provide support for ORBITER (after all it's free!)
Before posting a bug report make sure you have got the latest ORBITER release, and that your problem is not already documented in the FAQ or the bug forum.

3. Installation

3.1. Hardware Requirements

The standard ORBITER distribution requires the following hardware specs:

- 233MHz PC or better (Pentium, Athlon, etc.)
- 64MB system memory or more
- Windows 95/98/Me/NT/2000
- Direct3D 7.0 or higher, and DirectX compatible 3D graphics accelerator card with at least 4MB of video memory; 16bit z-buffer or higher (24bit or higher recommended)
- 640x480 display resolution (800x600 or higher recommended)
- Approximately 30MB of free disk space (mostly for planetary texture maps)
- DirectX compatible joystick (optional)

If you want to use the high resolution planetary texture packages, some additional requirements apply:

- AGP 3D graphics accelerator card with 32MB of video memory or more
- Hardware support for DXT1 texture compression (important!)
- 300MHz PC or better
- 128MB system memory or more
- Additional 16MB of disk space for each planetary texture.

Please read the section "Notes on high resolution textures" below before downloading the hires packages.

Since ORBITER keeps evolving these specs tend to become obsolete over time. If you don't get a reasonable frame rate (say >= 20fps) using the default Orbiter.cfg on a machine which meets the specs then please drop me a note and I will correct the requirement list upward.

3.2. Notes on DirectX

ORBITER uses Microsoft's 3D API, DirectX for interfacing with the 3D hardware. DirectX is not included in the ORBITER distribution. On most Windows systems DirectX will already be installed. If not, it can be downloaded from the Microsoft web site. You need version 7 to run ORBITER. Later versions should also work, but I have not tested them.

Please beware that no uninstall option is provided for DirectX. If at some point you need to remove DirectX for any reason you could face a complete re-installation of Windows.

3.3. Notes on High Resolution Textures

The ORBITER standard texture package contains planetary textures up to resolution level 7 (4096x2048 pixel bitmap size). The hires packages for Earth, the Moon and Mars contain texture maps for these planets up to resolution level 8 (8192x4096).

The hires textures require better than average graphics hardware. See "Hardware requirements" above for minimum specs.

PLEASE DO NOT DOWNLOAD THE HIRES TEXTURES IF YOUR GRAPHICS CARD DOES NOT SUPPORT DXT1 TEXTURE COMPRESSION! If in doubt please consult your card manufacturer's manual.

Without DXT1 support ORBITER must decompress all textures at the program start. This will take a very long time, and in addition it will increase memory requirements 8-fold. The textures up to level 8 for a single planet would in that case require 130MB of video memory!

Even if you think your hardware can handle the hires textures, you should download the standard version (*Base* and *Textures* package) first and make sure ORBITER runs fine, before downloading the hires packages.

If downloading the hires packages takes too long there is an alternative: Download the *Textool* package and "roll your own": get high-resolution planetary maps from the internet and use the pltex tool to convert them into ORBITER textures. See Section 17 for usage of pltex.

3.4. Download

The ORBITER distribution can be obtained from the Download page of the ORBITER site, http://www.medphys.ucl.ac.uk/~martins/orbit/orbit.html. The *Base* and *Textures* packages are required, other packages are optional. When upgrading you should only download packages with a time stamp more recent than the ones you have got already. In particular, there is no need to download the hires planetary texture packages more than once.

To minimise download times you should use a mirror closest to your location. Make sure to download the packages in binary mode. If you have problems downloading you may want to try at a different time of day.

3.5. First-time installation

- Create a directory for the ORBITER installation, e.g. \Program Files\Orbiter.
- Download the *Base* package and *Textures* package from the ORBITER Download page into this directory and unzip them. Take care to preserve the directory structure of the packages.
- Run Orbiter.exe and select suitable simulation and video parameters.

3.6. Upgrading from a previous installation

- It is recommended to back up your existing installation in case the new snapshot causes problems. At least you should back up any customised configuration files, meshes, textures, add-ons, etc.
- Download the *Base* package into your Orbiter directory and unpack, retaining the package directory structure. Answer "Yes" if prompted about overwriting files.

- Download and unpack the *Textures* package only if there is a newer version available.
- Run Orbiter.exe and select suitable simulation and video parameters.

3.7. Uninstall

- Remove the Orbiter directory with all contents and subdirectories. ORBITER does not modify the registry or any system resources.

4. The Launchpad

Starting Orbiter.exe brings up the *Orbiter Launchpad* dialog box. From here, you can

- set simulation, video and joystick parameters
- select a startup scenario
- open the online help system
- launch the Orbiter simulation window, or
- exit to the desktop.

4.1. Scenario tab

Figure 1: Launchpad dialog, Scenario tab

Scenario:
Contains a list of available scenarios. Select one and launch it with the “Orbiter” button. Below the list is a description of the currently selected scenario.

Special scenarios:

- The ***(Current state)*** scenario is automatically generated whenever you exit the simulator. Use this to continue from the latest exit state.
- The ***(Quicksave state)*** scenario contains the last in-game save state generated by pressing Ctrl-S.

Options:

- **Start paused**: Pause simulation on start. Press Ctrl+P to unpause after launch.

Save current:
Save the current exit state under a new name with a custom description.

4.2. Parameters tab

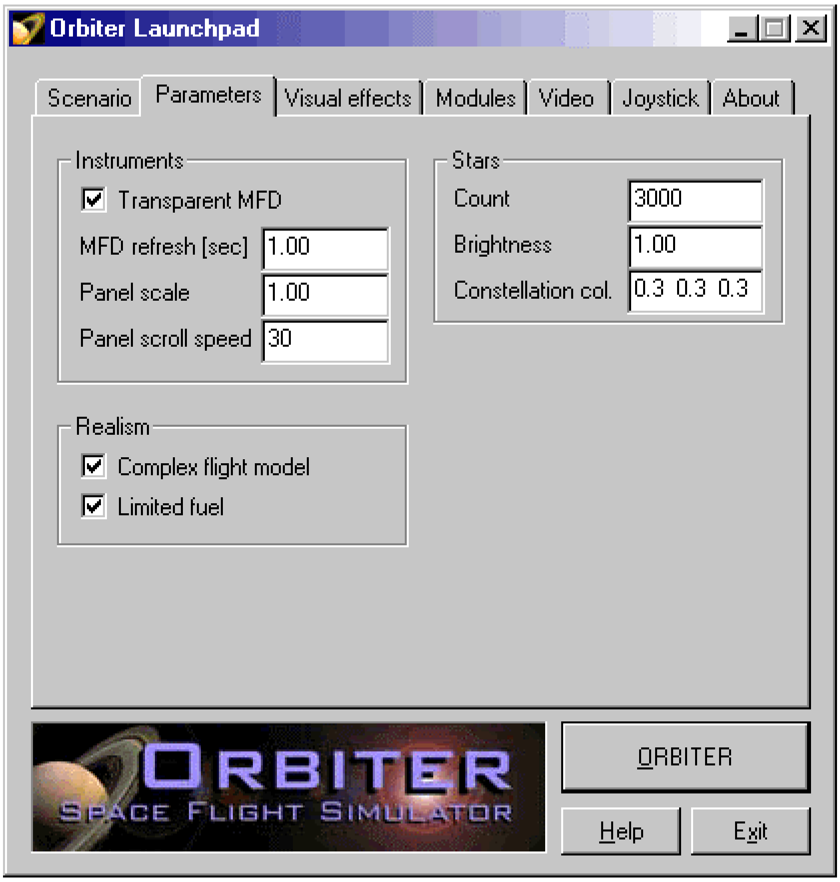


Figure 2: Launchpad dialog, Parameters tab

Instruments

- **Transparent MFD**: Make the onscreen multifunctional displays transparent. This provides a better view of the 3D environment, but makes it more difficult to read the instruments.
- **MFD refresh**: Time (in seconds) between MFD updates. Shorter intervals provide smoother updates, but may degrade performance.
- **Panel scale**: Sizing factor for instrument panels. Scale 1 provides optimal visual quality, but other values may be used to adapt the panel size at low or high screen resolutions.
- **Panel scroll speed**: Determines how fast the panel can be scrolled across the screen [pixels/second]. Negative values invert the panel scroll direction.

Stars

- **Count**: Number of displayed background stars. Orbiter contains a database of about 16000 bright stars. Specifying a large number will provide a more impressive night sky, but may degrade performance. Set to zero to suppress background stars.
- **Brightness**: Brightness scaling factor for background stars. Valid range is 0.2 - 2. Note that the dynamic range becomes less realistic for large values.
- **Constellation col**: Colour (RGB triplet) to use for drawing constellations (F9). Valid range for each value is 0 - 1.

Realism

- **Complex flight model**: Select the realism of the flight model for spacecraft.
- **Limited fuel**: Un-tick this box to ignore fuel consumption of your spacecraft.

Some of the more “realistic” spacecraft, such as the Space Shuttle, may *NOT* work properly if “Limited fuel” is not selected, because they rely on the reduction of mass during liftoff as a consequence of fuel consumption.

4.3. Visual effects tab

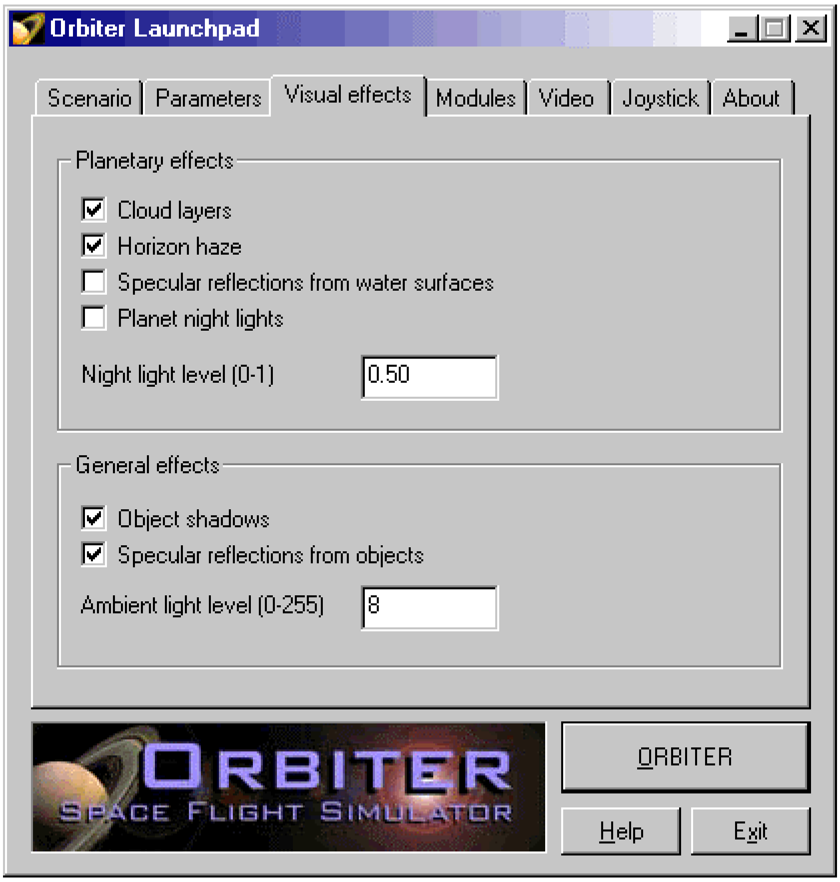


Figure 3: Launchpad dialog, Visual effects tab

Planetary effects

- **Cloud layers**: Render clouds as a separate mesh layer for appropriate planets. May degrade performance.
- **Horizon haze**: Render intensity-graded (“glowing”) horizon layer for planets with atmospheres. May degrade performance.
- **Specular reflections from water surfaces**: Render water surfaces on planets with specular reflection effects. May degrade performance.
- **Planet night lights**: Render city lights on the dark side of planet surfaces where available. May degrade performance.
- **Night light level**: Defines the brightness of night city lights. Valid range is 0 to 1. (ignored if planet night lights are disabled)

General effects

- **Object shadows**: Enable dynamic shadows of ground-based objects such as buildings.
- **Specular reflections from objects**: Render reflective surfaces like solar panels, window panes or metallic surfaces. May degrade performance.
- **Ambient light level**: Defines the brightness of the unlit side of planets and moons. Ambient level 0 is the most realistic, but makes it difficult to spot objects in the dark. Level 255 is uniform lighting (no darkness).

4.4. Modules tab

Figure 4: Launchpad dialog, Modules tab

This tab allows to activate and deactivate plugin modules for ORBITER which can extend the functionality of the core simulator. The modules provided with the standard ORBITER distribution are demos from the SDK package, and are available in full source code. Additional modules may be produced by add-on developers.

CustomMFD: This module provides an additional "Ascent MFD" mode for the multifunctional displays, which can be selected via Shift-P.

Rcontrol: remote control of ship engines. This allows to manipulate vessels even if they don't have input focus. Only runs in window mode.

Warpcontrol: allows fine-tuning of time acceleration. Only runs in window mode.

4.5. Video tab

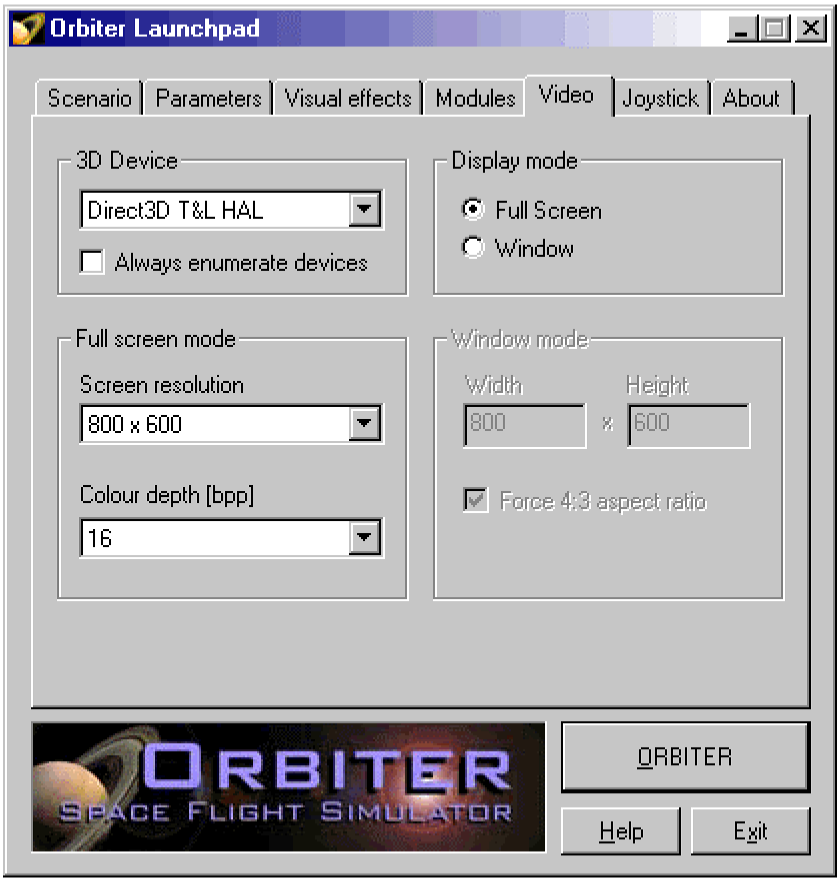


Figure 5: Launchpad dialog, Video tab

3D Device: Lists the available hardware and software devices for 3D rendering. Select a hardware device when possible, such as Direct3D HAL or Direct3D T&L HAL. Software devices such as RGB Emulation will produce poor performance. Note that some hardware devices do not support window mode.

Always enumerate devices: Tick this box if Orbiter does not display 3D devices or screen modes correctly. This option enforces a hardware scan whenever Orbiter is launched and skips the device data stored in device.dat. Make sure to tick this box after upgrading your graphics hardware or DirectX/video drivers to make Orbiter aware of the changes.

Display mode: Select fullscreen or windowed mode. Not all devices support windowed mode. Running in fullscreen mode may improve performance.

Fullscreen mode: Select screen resolution and colour depth for fullscreen mode. Only modes supported by the selected device are listed here. Higher resolution and colour depth will improve the visual appearance at the cost of reduced performance.

Window mode: Select the size of the render window for windowed modes. Select a width:height ratio close to 4:3 for best results. Large window sizes will reduce performance.

4.6. Joystick tab

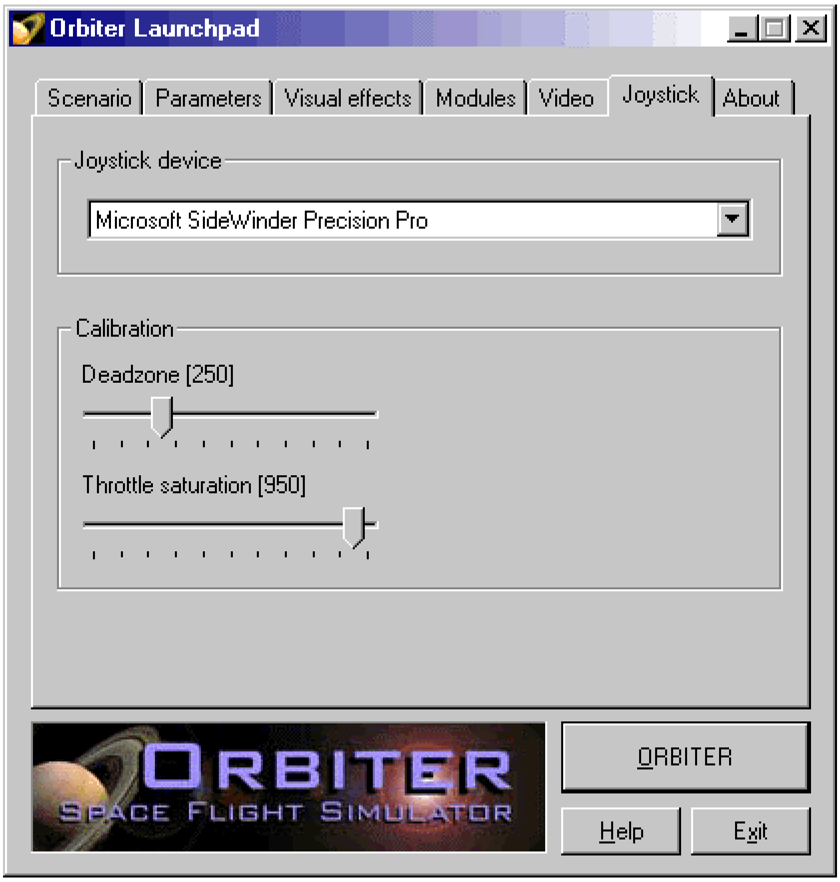


Figure 6: Launchpad dialog, Joystick tab

Joystick device: Lists all attached joysticks.

Deadzone: Use this to define how soon the joystick will respond when moved out of its centre position. Smaller values make it respond sooner. Increase if attitude thrusters do not cut out completely in neutral position.

Throttle saturation: Defines the tolerance zone at the minimum and maximum range of the throttle control at which the joystick reports zero and maximum throttle, respectively. Reduce if main engines do not cut out completely at minimum throttle setting. (Applies only to joysticks with throttle control).

If further calibration is required you should use the appropriate tools in the Windows Control Panel.

5. Keyboard Interface

The key assignment reference in this section and the rest of the manual refers to the keyboard layout shown in Figure 7. For other layouts (e.g. language-specific) the key labels may be different. The relevant criterion for key functions in Orbiter is the *position of the key* on the keyboard, not the key label. For example, on the German keyboard, the keys for the “turn orbit-normal” (`;`) and “turn orbit-antinormal” (`'`) will be “Ö” and “Ä”.

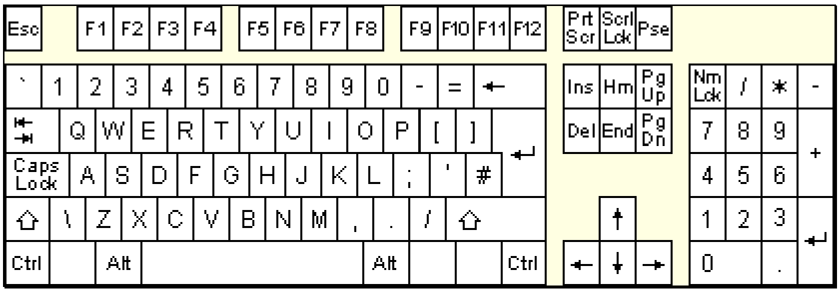


Figure 7: Keyboard layout reference

5.1. General

F	Toggle between frame rate and field-of-view display.
R	Time warp: Slow down simulation by factor 10 (down to real-time).
T	Time warp: Speed up simulation by factor 10 (up to a maximum warp factor of 1000).
X	Zoom out (increase field of view).
Z	Zoom in (decrease field of view).
Ctrl C	Comms: Request takeoff or landing clearance if within radio range of a space traffic control (STC) center.
Ctrl D	Comms: Request docking or undocking clearance if within radio range of the selected docking target (orbital station).
Ctrl P	Pause/resume simulation.
Ctrl Q	Exit to Launchpad dialog.
Ctrl S	Quicksave scenario.
F1	Toggle internal/external view of the user-controlled spacecraft.
Ctrl F1	Display selection list for external view targets.
F2	Toggle tracking mode for external camera views (target-relative / absolute direction / global frame).
Ctrl F2	Display selection list for camera tracking modes (external views only).
F3	When the user-controlled ship is parked on a launch pad or docked to an orbital station, this allows to switch to any ship currently landed at the same spaceport or docked to the same station.
Ctrl F4	Main menu.
F8	In cockpit view, toggle between onscreen instruments and custom panels (if supported by the spacecraft).
F9	“Planetarium mode”: Toggle display of constellations.

5.2. Spacecraft Controls

These keys allow manual maneuvering of the user-controlled spacecraft. See also joystick controls.

Note that manual control of the attitude thrusters (Numpad 1/2/3/4/6/8 and corresponding joystick controls) is disabled in accelerated time mode.

Main/retro thruster controls:

Ctrl + *Numpad*	Accelerate by increasing main thruster setting or by decreasing retro

	thruster setting.
Ctrl - Numpad	Decelerate by decreasing main thruster setting or by increasing retro thruster setting.
* Numpad	Kill main and retro thrusters.
+ Numpad	Fire main thrusters at 100% while pressed (overrides permanent setting)
- Numpad	Fire retro thrusters at 100% while pressed (overrides permanent setting)

Note that not all craft support retro thruster controls.

Hover thruster controls (where available):

0 Numpad	Increase hover thruster setting.
. Numpad	Decrease hover thruster setting.

Note that not all craft support hover thrusters.

Attitude thruster controls (rotational mode):

4 / 6 Numpad	Engage attitude thrusters for rotation around longitudinal axis (bank)
2 / 8 Numpad	Engage attitude thrusters for rotation around transversal axis (pitch)
1 / 3 Numpad	Rotational mode: Engage attitude thrusters for rotation around vertical axis (yaw)
5 Numpad	Toggle “Kill rotation” navigation computer mode. Stops spacecraft rotation by engaging appropriate attitude thrusters

Note: In combination with Ctrl, thrusters are engaged at 10% max. thrust for fine control.

Attitude thruster controls (linear mode):

2 / 8 Numpad	Engage attitude thrusters for up/down translation.
1 / 3 Numpad	Engage attitude thrusters for left/right translation.
6 / 9 Numpad	Engage attitude thrusters for forward/back translation

Note: In combination with Ctrl, thrusters are engaged at 10% max. thrust for fine control.

Other controls:

/ Numpad	Toggle attitude thruster mode between rotational (engage opposite thruster pairs) and linear (engage parallel thruster pairs)
A	Toggle “Hold altitude” navcomp mode. Maintain current altitude above surface by means of hover thrusters only. This will fail if hover thrusters cannot compensate for gravitation, in particular at high bank angles. Combining this mode with the “H-level” mode is therefore useful.
L	Toggle “H-level” navcomp mode. This mode keeps the spacecraft level with the horizon by engaging appropriate attitude thrusters.
[	Toggle “Turn prograde” navcomp mode. This mode turns the spacecraft into its orbital velocity vector.
]	Toggle “Turn retrograde” navcomp mode. This mode turns the spacecraft into its negative orbital velocity vector.
;	Toggle “Turn orbit-normal” navcomp mode. Rotates spacecraft normal to its orbital plane (in the direction of $\vec{R}\times\vec{V}$)
'	Toggle “Turn orbit-antinormal” navcomp mode. Rotates spacecraft antinormal to its orbital plane (in the direction of $-\vec{R}\times\vec{V}$)
Insert / Delete Cursorpad	Trim control (only vessels with aerodynamic surfaces)

5.3. External Views

I	Toggle display of information about current object.
Page Up	Move camera away from target object.
Page Down	Move camera towards target object.
Ctrl ↑ ↓ → ←	Rotate camera around object.

5.4. Internal (cockpit) view

The two multifunctional displays (MFD) on the left and right side of the screen are controlled via left/right Shift key combinations, where the left Shift key addresses the left MFD, the right shift key addresses the right MFD.
The Head-up display (HUD) and MFDs are visible only in internal cockpit view.

F8	Map between onscreen MFD mode and panel mode (if available)
↑ ↓ → ←	Scroll instrument panel.
Ctrl ↑ ↓ → ←	Switch to neighbour panel, if available.
Ctrl H	Toggle HUD display on/off.
H	Switch HUD mode.
Ctrl M	Open an input box for HUD reference selection (Orbit and Docking HUD only)
Shift F1	Open a menu for left/right MFD mode selection.
Shift ‘	Open/page/close the MFD-specific parameter selection menu.
Shift A	Display *Align orbital plane* mode MFD.
Shift D	Display *Docking* mode MFD.
Shift L	Display *Launch/landing* mode MFD.
Shift M	Display *Map* mode MFD.
Shift O	Display *Orbit* mode MFD.
Shift S	Display *Surface* mode MFD.
Shift X	Display *Transfer* mode MFD.
Shift V	Display *Synchronise orbit* mode MFD.
Shift Q	Turn off MFD.

Docking Mode MFD

Shift T	Input new docking target.

Align orbital plane Mode MFD

Shift T	Input new target object.

Orbit Mode MFD

Shift A	Auto-select reference object.
Shift M	Toggle display mode (list only, graphics only and both)

Shift N	No target orbit.
Shift P	Toggle orbit projection mode (ecliptic, ship's and target's orbital plane)
Shift R	Select new reference object (planet or moon) for orbit calculation.
Shift T	Open menu for target selection.

Map Mode MFD

Shift R	Open input box for reference planet/moon selection.
Shift T	Open a menu for target selection.

Transfer Mode MFD

Shift R	Open input box for reference planet/moon selection.
Shift S	Open a menu for source orbit object selection.
Shift T	Open a menu for target selection.
Shift N	Unselect target.
Shift X	Toggle (hypothetical) transfer orbit display on/off.
Shift M	Toggle numerical multibody trajectory calculation.
Shift U	Refresh numerical trajectory, if displayed.
Shift Z	Open input box for time step definition.
Shift , / .	Rotate transfer orbit ejection longitude.
Shift - / =	Decrease/increase transfer orbit major axis.

Synchronise orbit Mode MFD

Shift T	Input new target object (orbital station or moon)
Shift M	Change reference axis.
Shift N	Select number of entries in the reference transit time list.
Shift , / .	Rotate reference axis (in manual mode only).

5.5. Menu Selections

↑	Move to previous item in the list.
↓	Move to next item in the list.
→	Display sub-list for selected item, if available.
←	Go back to the parent list from a sub-list.
Enter	Select current item and close list.
Esc	Cancel list.

6. Joystick Interface

A joystick can be used to operate the attitude and main thrusters of the user-controlled spacecraft manually.

Action	Effect
Push stick left or right	Rotate around vessel's longitudinal axis (bank)

Push stick forward or backward	Rotate around vessel's transversal axis (pitch)
Operate rudder control *or* Push stick left or right while holding joystick button 2	Rotate around vessel's vertical axis (yaw)
Operate throttle control	Controls main thruster settings. This is similar to the Ctrl + *Numpad* and Ctrl - *Numpad* keyboard controls, but it affects only the main thrusters, not the retro thrusters.

If the joystick has a direction controller ("coolie hat") it can be used to rotate the camera around the observed object in external view mode.

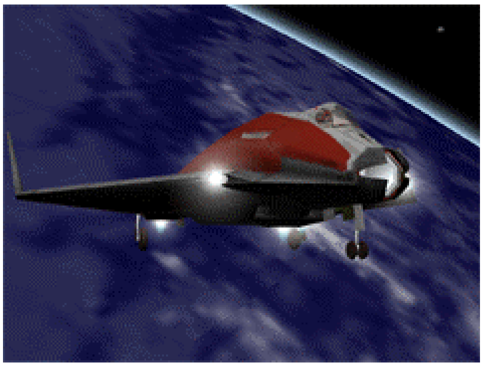
7. Mouse interface

Spacecraft instrument panels can be operated by the mouse. Most buttons, switches and dials are activated by pressing the left mouse button. Some elements like multi-way dials may respond to both left and right mouse buttons.

8. Spacecraft classes

The following standard spacecraft types are currently available in Orbiter. Many more can be downloaded as add-ons. See the Orbiter web site for a list of add-on repositories.

8.1. Delta-Glider



This is the new Mk3 Delta-Glider designed by Roger "Frying Tiger" Long. The new version comes with operating landing gear, nose cone docking port and airlock door.

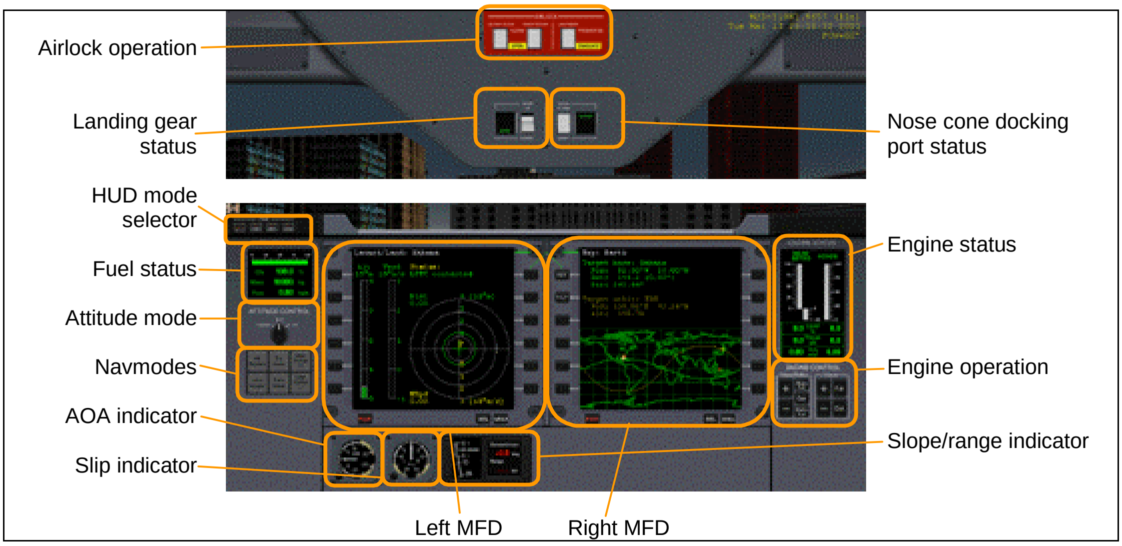
The Mk3 is also the first vessel to make use of the new instrument panels.

The Delta Glider is the ideal ship for the novice pilot to get spaceborne. Its futuristic design concept and extremely low fuel consumption make it easy to achieve orbit, and it can even be used for interplanetary travel. The winged design provides aircraft-like handling in the lower atmosphere, while the vertically mounted hover-thrusters allow vertical takeoffs and landings independent of atmospheric conditions and runways.
The Mk3 has a docking port and airlock under the nose cone.

Technical specifications:

Mass	$8{\cdot}10^3$ kg	(no fuel)
	$18{\cdot}10^3$ kg	(100% fuel)
Length	17.76 m	
Wingspan	17.86 m	
Thrust	$2.16{\cdot}10^5$ N	(2 main thrusters)
	$5.4{\cdot}10^4$ N	(2 wing-mounted retro thrusters)
	$2.16{\cdot}10^5$ N	(3 hover thrusters)
Isp	$5{\cdot}10^4$ m/s	(fuel-specific impulse)
Inertia (PMI)	15.5 / 22.1 / 7.7 m^2	
Stall C_L	1.0	
Stall AOA	20°	

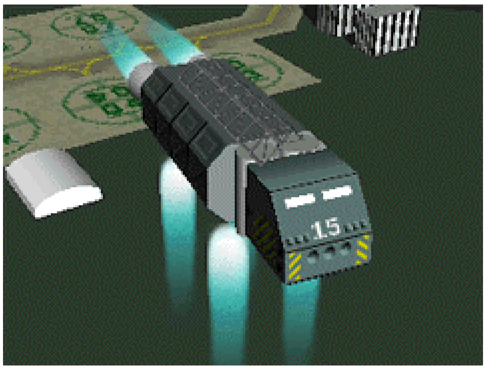
Main and overhead instrument panels:



Vessel-specific key controls:

G	Operate landing gear
K	Operate nose cone docking mechanism
O	Open/close outer airlock door

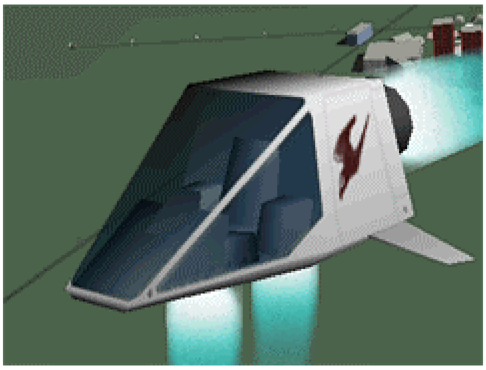
8.2. Shuttle A



Mass:	$6\cdot10^3$ kg (empty), $18\cdot10^3$ kg (full)
Length:	20 m
Thrusters:	$2.7\cdot10^5$ N (main) $1.0\cdot10^5$ N (retro) $1.89\cdot10^5$ N (hover)

Does not produce lift during atmospheric flight.

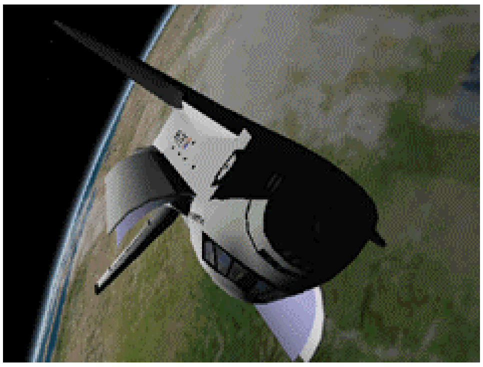
8.3. Shuttle PB (PTV)



Mass:	$2\cdot10^3$ kg (empty), $5\cdot10^3$ kg (full)
Length:	7 m
Thrusters:	$1.2\cdot10^5$ N (main) $4.0\cdot10^4$ N (retro) $8.0\cdot10^4$ N (hover)

Very agile single-seater (based on Balázs Patyi's PTV model). Produces little lift in atmospheric flight.

8.4. Space Shuttle Atlantis



The first “real” spacecraft to appear in the Orbiter standard distribution. Launch configuration consists of Orbiter, main tank and 2 solid rocket boosters (SRB). 3D model by Javier Fernandez.

Technical specifications:

Orbiter		
Length	39.16 m	
Wingspan	24.54 m	
Height	14.29 m	
Mass	2041166 kg	(liftoff)
	104326 kg	(end of mission)
Max. cargo	28803 kg	

Tank[b]		
Length	47.83 m	
Diameter	9.68 m	
Mass	35425 kg	(empty)
	719115 kg	(propellant)
	756445 kg	(total)

SRB		
Length	45.7 m	
Diameter	3.8 m	(tube)
	5.9 m	(max)
Mass	87543 kg	(empty)
	502126 kg	(propellant)
	589670 kg	(total)
Thrust	11791820 N	(liftoff)

Orbiter+Tank assembly	
Length	57.55 m
Height	24.44 m

Orbiter+Tank+SRBs (launch assembly)	
Length	57.91 m
Height	24.44 m

[a]: principal moments of inertia tensor, mass-normalised, assuming homogeneous density distribution

[b]: including Orbiter mount brackets

Atlantis-specific key controls:

J	Jettison: separate SRBs or main tank
K	Operate cargo bay doors
G	Operate landing gear (activated only after tank separation)

Launch:

- Fire main engines at 100%.

- SRBs are ignited automatically when main engines reach 95%. SRBs are not controlled manually. Once ignited, they cannot be shut off.
- During launch, attitude is controlled via SRB thrust vectoring. Roll shuttle for required heading, and decrease pitch during ascent for required orbit insertion.
- SRBs separate automatically at T+2:06min. In an emergency, SRBs can be jettisoned manually with J.
- Ascent continues with Orbiter main engines. Throttle down as required for 3g max acceleration.
- Tank separates at T+8:58min (alt 110km) when empty, or manually with J.
- After tank separation, orbiter switches to OMS (orbital maneuvering system) using internal tanks, for final orbit insertion. Attitude thrusters (RCS – reaction control system) are activated.

Docking:
- The orbiter carries a docking attachment in the cargo bay.
- Open cargo bay doors before docking.
- Docking direction is in orbiter's +y direction (up). The Docking MFD must be interpreted accordingly.

Unlike the futuristic spacecraft designs, Atlantis provides only a small margin of error for achieving orbit. Try some of the other ships before attempting to launch the Shuttle. Limited fuel *must* be selected, otherwise Atlantis is too heavy to reach orbit!

9. Camera modes

9.1. Internal view

In internal (cockpit) view the player is placed inside the cockpit of his/her spaceship and looks forward. Head-up display (HUD) and multifunctional displays (MFD) are only displayed in internal view. To return to cockpit view from any external views, press F1.

9.2. External views

External views allow to have a look at any objects currently populating the simulated solar system, including the Sun, planets and moons, spacecraft, orbital stations and surface bases. To bring up a list of objects to look at, press Ctrl F1. See Section 5.5 on how to navigate selection lists.

Figure 8: Solar system main selection list.

In external view a display of target parameters can be toggled by pressing I.
From cockpit view, an external view of the current spaceship can be selected by pressing F1.
In movable external views the camera can be rotated around the target object by pressing Ctrl ↓ ↑ → ← keys. The Page Up and Page Down keys move the camera towards or away from the target.

Different camera panning modes for external views can be selected by pressing F2 or via a menu by pressing Ctrl F2:

- **Target-relative**: The camera is fixed in the target's local frame of rotation. Looking at a planet in this mode for example will rotate the camera together with the planet around its axis. Ctrl ↓↑→← will rotate the camera around the target's local axes.
- **Global frame**: The camera is fixed in a non-rotating reference frame. Looking at a planet in this mode will show the planet rotating underneath the camera. Ctrl ↓↑→← will rotate the camera around the axes of the ecliptic frame of reference.
- **Absolute direction**: This can be regarded as a mixture of the two modes above: The direction into which the camera points is fixed in an absolute frame, but it is tilted with respect to the target's local frame. Ctrl ↓↑→← will rotate the camera around the target's local axes.
- **Target to** …: Positions camera so that the specified object is behind the target.
- **Target from** …: Positions camera so the specified object is behind the camera.

In *Target to …* and *Target from …* modes camera rotation (Ctrl ↓↑→←) is deactivated, but radial camera movement with Page Up and Page Down is still available.

10. Head-up Display

The Head-up display (HUD) provides information about the current status of your spacecraft when internal (cockpit) view mode is selected. The HUD is switched on/off with Ctrl H. HUD modes can be selected with H. The following modes are available:

- **Surface**: Displays horizon pitch ladder, compass ribbon, altitude and "airspeed".
- **Orbit**: Displays orbital plane pitch ladder, prograde and retrograde velocity markers
- **Docking**: Displays target distance and relative velocity markers.

All HUD modes show engine and fuel status in the top left corner, and general information (time and camera aperture) in the top right corner. Two multifunctional displays (MFDs) can be displayed independent of the HUD mode (see Section 11).

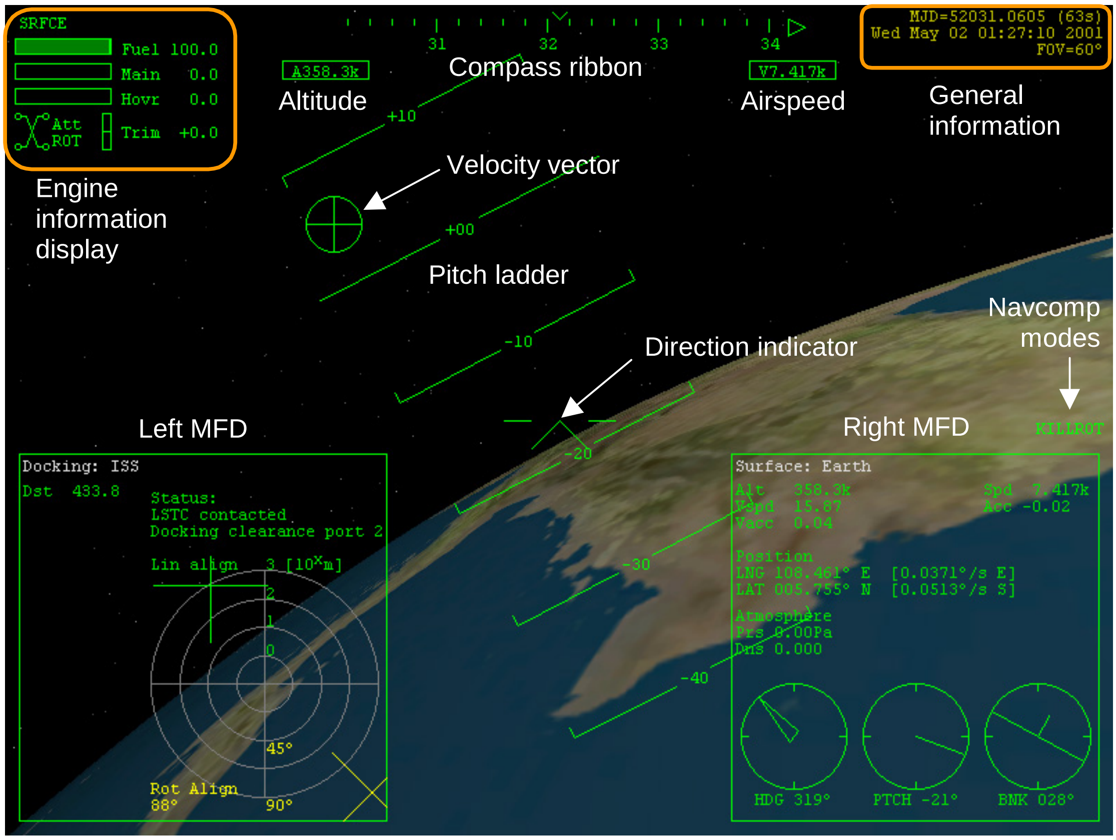


Figure 9: HUD in surface mode, including docking and surface MFDs.

10.1. General information display

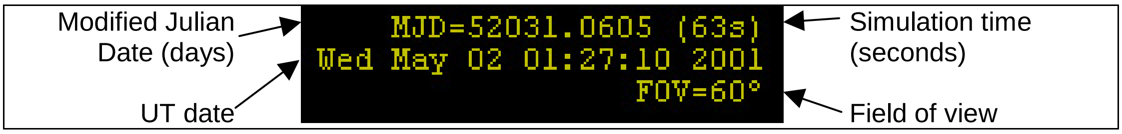


Figure 10: General simulation information

MJD: The *Julian Date* (JD) is the interval of time in mean solar days elapsed since 4713 BC January 1 at Greenwich mean noon. The *Modified Julian Date* (MJD) is the Julian Date minus 240 0000.5
UT: Universal time counted from 0^h at midnight. Unit is mean solar day.
FOV: (vertical) field of view, i.e. viewport camera aperture.

The display can be toggled between FOV and viewport resolution/frame rate via the F key.

10.2. Engine information display

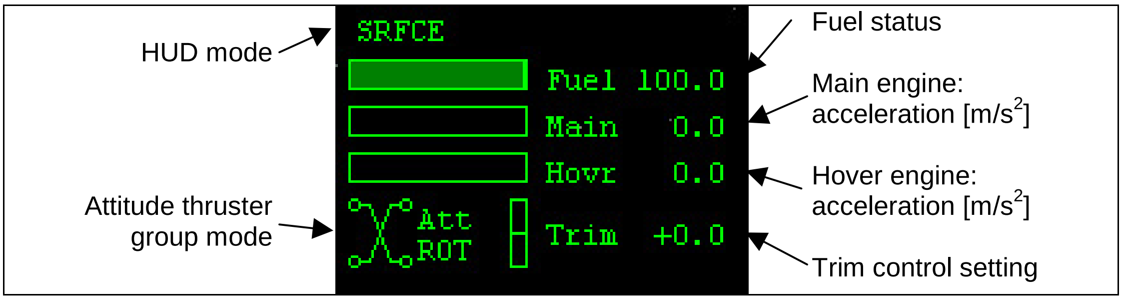


Figure 11: Fuel/engine displays

The engine information display is only shown in non-panel views.

Fuel status: Remaining fuel is displayed as percentage of full tanks.
Main engine: The horizontal bar shows current main/retro engine thrust as fraction of max. engine thrust. Green indicates main thrusters, orange indicates retro thrusters. The numerical value shows acceleration in units of m/s^2 (positive for main, negative for retro thrust). Note that the acceleration may change even if the thrust setting doesn't, because the ship's mass changes as fuel is consumed.
Hover engine: If available, hover engines are mounted underneath the ship's fuselage to assist in surface flight, in particular during takeoff/landing. Display analogous to main engine.
Attitude thruster mode: Attitude thrusters are small engines used for rotation and fine translational adjustments. The display shows the current mode (rotational/translational).
Trim setting: Displays the current setting of the trim control (if available). Trimming allows to adjust the flight characteristics during atmospheric flight.

For more information about engines and spacecraft control see Section 13.

10.3. Surface Mode

Indicated by "SRFCE" in the upper left corner.
This mode displays a pitch ladder which indicates the ship's orientation w.r.t. the current plane of the horizon. The plane of the horizon is defined by its normal vector, from the planet centre to the spacecraft.
The compass ribbon at the top of the screen indicates the ship's forward direction w.r.t. geometric north. A marker shows the direction of the current target (spaceport).
The box left below the compass ribbon shows the current altitude [m]. The box right below the compass ribbon shows the current "airspeed" [m/s] (even if there is no atmosphere).
The surface-relative velocity vector direction is marked by "⊕".

10.4. Orbit Mode

Indicated by "ORBIT *Ref*" in the upper left corner, where *Ref* is the name of the reference object.
This mode displays a pitch ladder relative to the current orbital plane, where the "0" line indicates the orbital plane. It also marks the direction of the orbital velocity vector (prograde direction) by "⊕ " and retrograde direction by "+". If neither the prograde nor retrograde direction is visible, then the direction of the ⊕ marker is indicated by a pointer labeled "PG" (prograde).
The reference object for the HUD can be manually selected by pressing Ctrl M.

10.5. Docking Mode

Indicated by "DOCK *Tgt*" in the upper left corner, where *Tgt* is the name of the target station.
This mode marks the current docking target (orbital station) with a square marker, and displays its name and distance. It also shows the direction and magnitude of the target-relative velocity vector. The velocity of the target relative to the ship is indicated by "⊕ ". This is the direction in which you need to accelerate to synchronise your speed with the target. The opposite direction (the velocity of the ship relative to the target) is indicated by "+". If neither ⊕ nor + are visible, then the direction of the ⊕ marker is indicated by a pointer. Similarly, if the target marker is offscreen, its direction is indicated by a pointer.
The target station for the HUD can be manually selected by pressing Ctrl M.

11. Multifunctional Display Modes

Two multifunctional displays (MFD) can be displayed simultaneously on the screen. MFDs provide essential information about all spacecraft flight parameters. A range of different MFD modes is available to assist in various navigational problems such as launch/landing, surface flight, orbit determination etc. MFDs are controlled by Shift key combinations, where the left Shift key controls the left MFD, the right Shift key controls the right MFD.
MFD modes can be selected either directly (see section 5.4 for keyboard commands) or by pressing Shift F1 to bring up a menu.
Each MFD mode may support specific parameters which can be selected with the appropriate Shift key combination. You don't need to remember the key codes, since MFDs which allow parameter selection provide a menu mode which can be selected with Shift ‘. Pressing Shift ‘ again will switch to the next menu page or return to the MFD display. To select a menu item, press Shift with the highlighted character key.

11.1. Orbit

The *Orbit* MFD mode displays a list of elements and parameters which characterise the ship's orbit around a central body, as well as a graphical representation. In addition, a target object (ship orbital station or moon) orbiting the same central body can be selected, whose orbital parameters will then be displayed as well.
Shift T opens a menu to specify a target object. Only targets which orbit around the current reference object will be accepted. The target display can be turned off with Shift N.

The plane into which the orbits are projected can be selected via Shift P. The current projection plane is indicated in the top right corner of the instrument. *Ecliptic* projects into the plane of the ecliptic (Earth's orbital plane). *Ship* projects into the vessel's current orbital plane, and *Target* projects into the target's current orbital plane, if a target is specified.

Key options:

Shift A	Auto-select reference object.
Shift M	Toggle display mode (list only, graphics only and both)
Shift N	No target orbit.
Shift P	Toggle orbit projection mode (ecliptic, ship's and target's orbital plane)
Shift R	Select new reference object (planet or moon) for orbit calculation.

Shift T	Open menu for target selection.

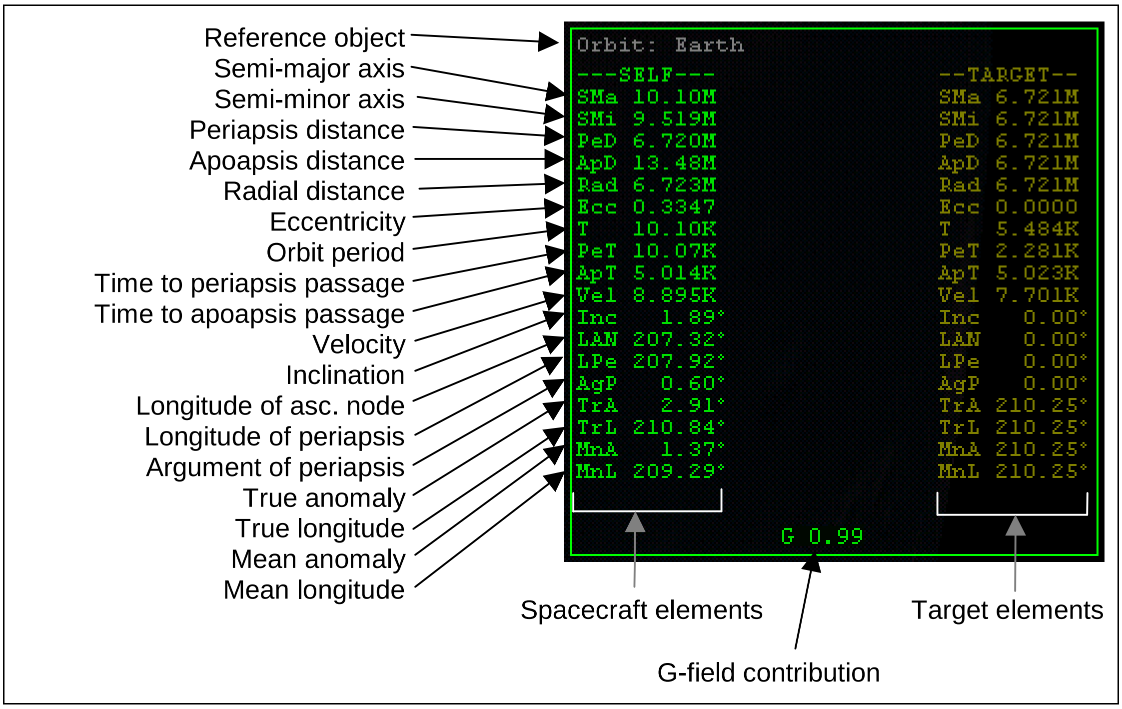


Figure 12: Orbit MFD in parameter list mode.

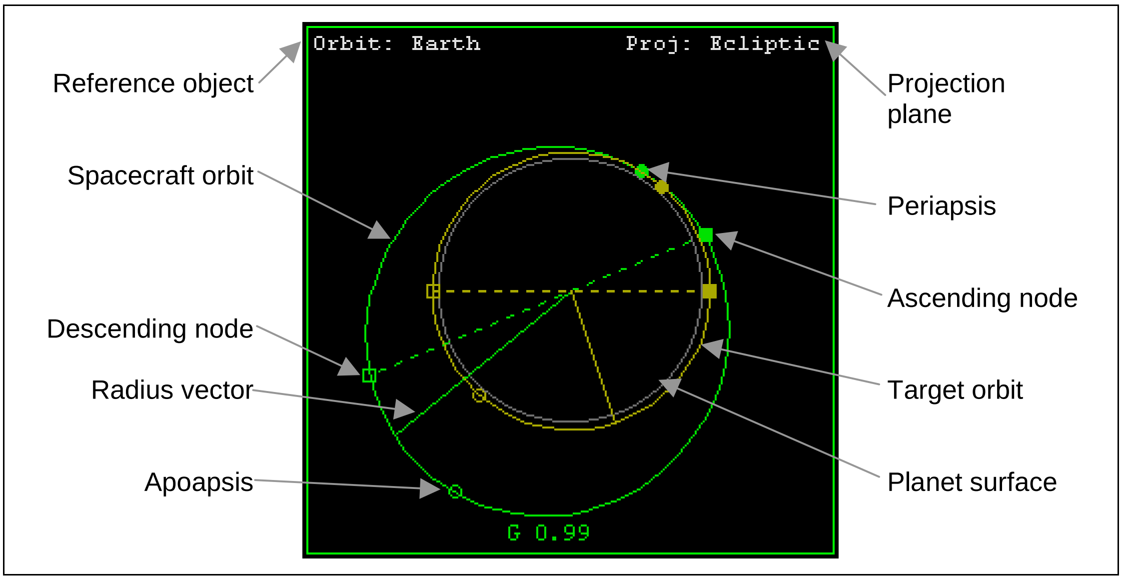


Figure 13: Orbit MFD in graphic mode.

<u>Notation:</u>

- Periapsis: The point of the orbital ellipse closest to the central body.
- Apoapsis: The point furthest away from the central body.
- Ascending node: The point at which the orbit passes through the ecliptic (or other reference plane) from below.
- Descending node: The point at which the orbit passes through the ecliptic from above.
- Radius vector: The vector from the central body to the current position of the orbiting body.

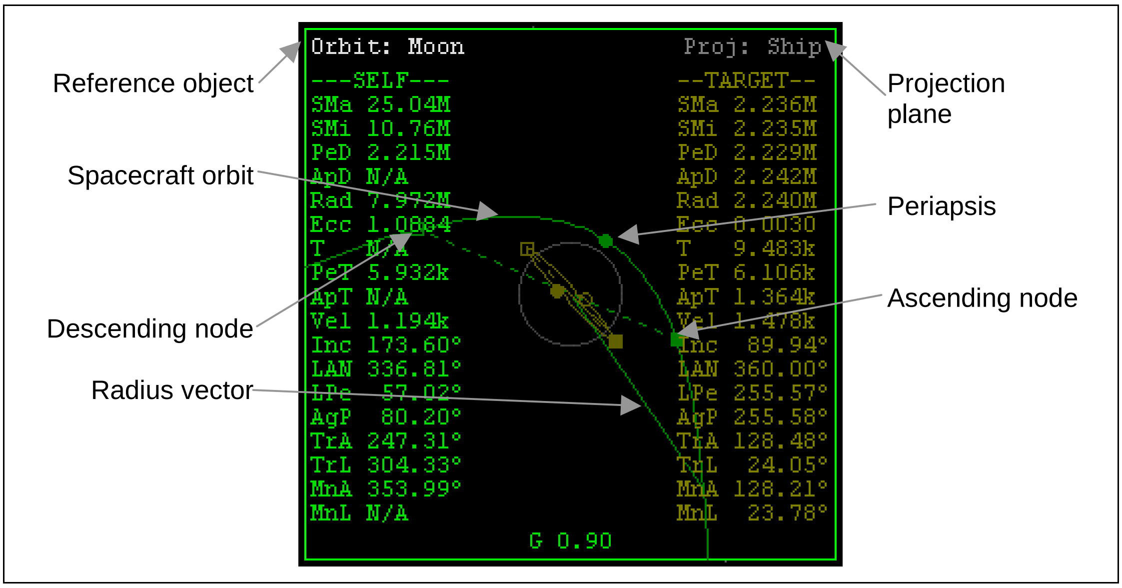


Figure 14: Hyperbolic orbit around the Moon

For hyperbolic orbits, the following parameters are interpreted specially:

SMa: real semi-axis *a*: distance from coordinate origin (defined by intersection of hyperbola asymptotes) to periapsis
SMi: imaginary semi-axis $b = a\, sqrt(1-e^2)$
ApD: apoapsis distance: not applicable
T: orbital period: not applicable
PeT: time to periapsis passage; negative after periapsis passage
ApT: time to apoapsis passage: not applicable
MnA: mean anomaly, defined as $e \sinh H - H$, with *H* hyperbolic eccentric anomaly

<u>G-field contribution</u>

The “G” value at the bottom of the display shows the relative contribution of the current reference body to the total gravity field at the ship’s position. This can be used to estimate the reliability of the Keplerian (2-body) orbit calculation. For values close to 1 a 2-body approximation is accurate. For low values the true orbit will deviate from the analytic calculation, resulting in a change of the orbital elements over time.
As a warning indicator, the G display will turn yellow for contributions < 0.8, and red if the selected reference object is not the dominant contributor to the gravity field. In that case, Shift A will select the dominant object.

For further explanation of orbital elements see Appendix C.

11.2. Landing

The *Landing* MFD mode contains graphical indicators for horizontal and vertical airspeed components, altitude, landing target direction and distance, and takeoff/landing clearance status.

- The altitude bar has a range from 1 to 10^4m (logarithmic scale).
- The vertical speed bar has a range from ±0.1 to $\pm 10^3$m/s (logarithmic scale). Positive vertical speed is indicated by a green bar, negative vertical speed by a yellow or red bar.
- Horizontal speed and target indicator are displayed relative to the ship's orientation, on a logarithmic radial scale. Hspeed range between 0.1 and 10^3m/s, target distance range between 1 and 10^4m.
- The landing cone for the current altitude is shown as a green or red circle. Green indicates that the ship is within the cone. The radius of the circle is a linear function of the ship's altitude.
- The status indicator shows if contact to Local Space Traffic Control is established, and the takeoff/landing clearance status.

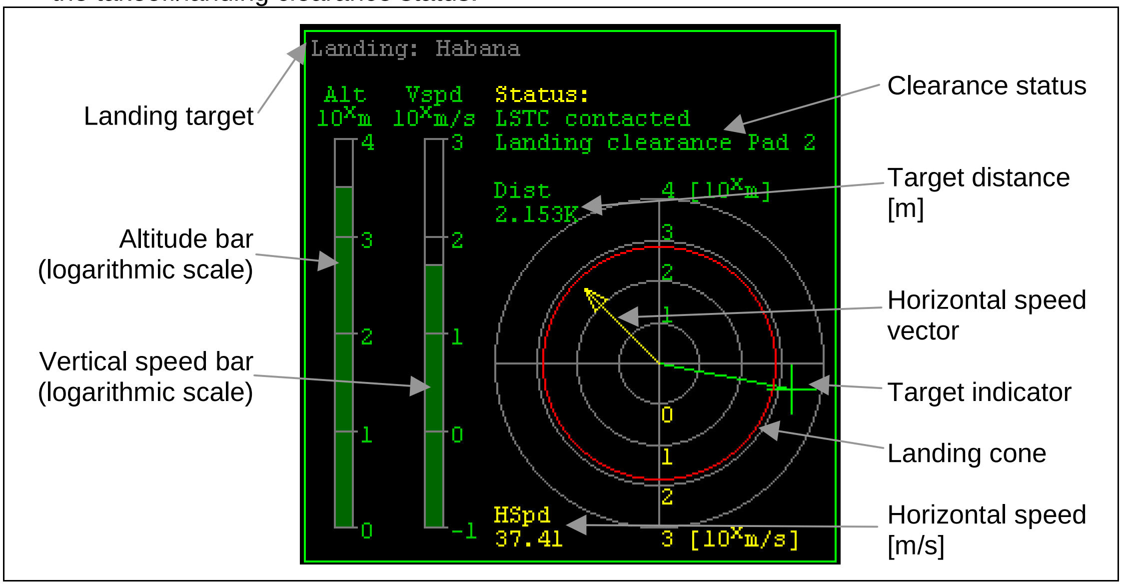


Figure 15: Landing MFD.

11.3. Docking

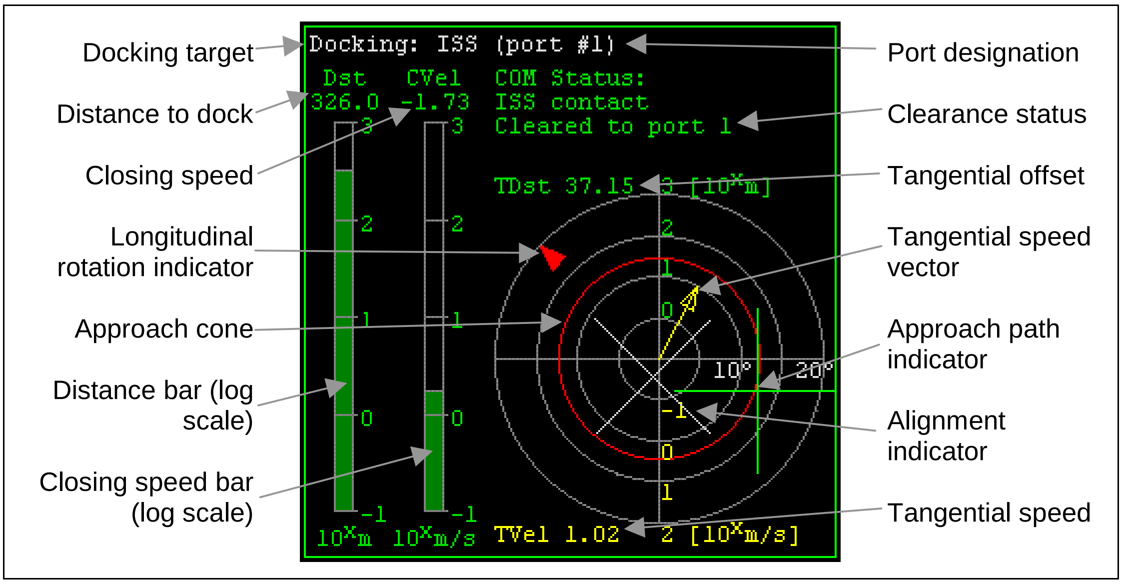


Figure 16: Docking MFD.

Key options:

Shift T	Input a new orbital station as docking target.

The *Docking* MFD assists during final approach to an orbital station. The layout is similar to the Landing MFD.

- **Docking target**: the currently targeted orbital station.
- **Port designation**: If a dock has been allocated for your ship, its designation is displayed in the title line.
- **Dst**: Distance to dock [m], or distance to station if no dock is allocated. The bar shows the distance on a logarithmic scale in the range 0.1–10^3 m.
- **CVel**: Closing speed [m/s]. The bar shows the closing speed on a logarithmic scale in the range 0.1–10^3 m/s. Yellow indicates positive closing speed.
- **COM status**: indicates whether the target is within radio range, and the current docking/undocking clearance status.

The circular instrument shows the ship's alignment with respect to the approach path towards the allocated dock.

- **TDst**: The *green cross* indicates the position of the approach path relative to the ship. When centered, the ship is aligned on the approach path. The radial scale is logarithmic in the range 0.1–10^3 m. The numerical value is the ship's tangential distance from the approach path [m]. Tangential alignment should be performed with attitude thrusters in *linear* mode (see Section 13.2).
- **TVel**: The *yellow arrow* indicates the relative tangential velocity of your vessel with respect to the target. The radial scale is logarithmic in the range 0.01–10^2 m/s. The numerical value is the tangential velocity [m/s]. *To align your ship with the approach path, engage linear attitude thrusters so that the arrow points towards the approach path indicator.*
- **Alignment indicator**: The *white cross* indicates the alignment of the ship's forward direction with the approach path direction. When centered, the ship's forward direction is parallel to the approach path. The cross turns red if misalignment is > 2.5° . The radial scale is linear in the range 0–20° . Rotational alignment should be performed with attitude thrusters in *rotational* mode (see Section 13.2).

- **Longitudinal rotation indicator**: This arrow indicates the ship's longitudinal alignment with the docking port. To align, the indicator must be moved into 12 o'clock position by rotating the ship around its longitudinal axis, by engaging bank attitude thrusters in rotational mode (see Section 13.2). When alignment is achieved, the indicator turns white (misalignment < 2.5°). Note that this indicator is only displayed when directional alignment (see above) is within 5° .
- **Approach cone**: The concentric red or green circle indicates the size of the approach cone at the current dock distance. The ship should approach the dock so that the approach path indicator is always inside the approach cone (indicated by a green circle). The approach cone becomes smaller as the ship approaches the dock.

Closing speed should be reduced as the ship approaches the dock (using retro thrusters). The final speed should be < 0.1 m/s.

You can only dock at a port allocated to you by the station's traffic controller. To request docking (and undocking), press Ctrl D. Clearance status is displayed in the MFD.

Note:
- Engine controls are disabled while docked to an orbital station.
- To dock successfully, you must approach the dock to within 0.3 m. Additional restrictions may be implemented in the future (speed, alignment, etc.)
- No collision checks are currently performed. If you fail to dock and keep closing in, you may fly your ship through the station.
- After docking successfully your ship will automatically be refueled.

11.4. Surface

The *Surface* MFD mode assists in flight close to planetary surfaces. It lists altitude, horizontal and vertical speed and acceleration, atmospheric pressure and density (if applicable), vessel position in equatorial coordinates, and indicators for heading, pitch and bank angles.

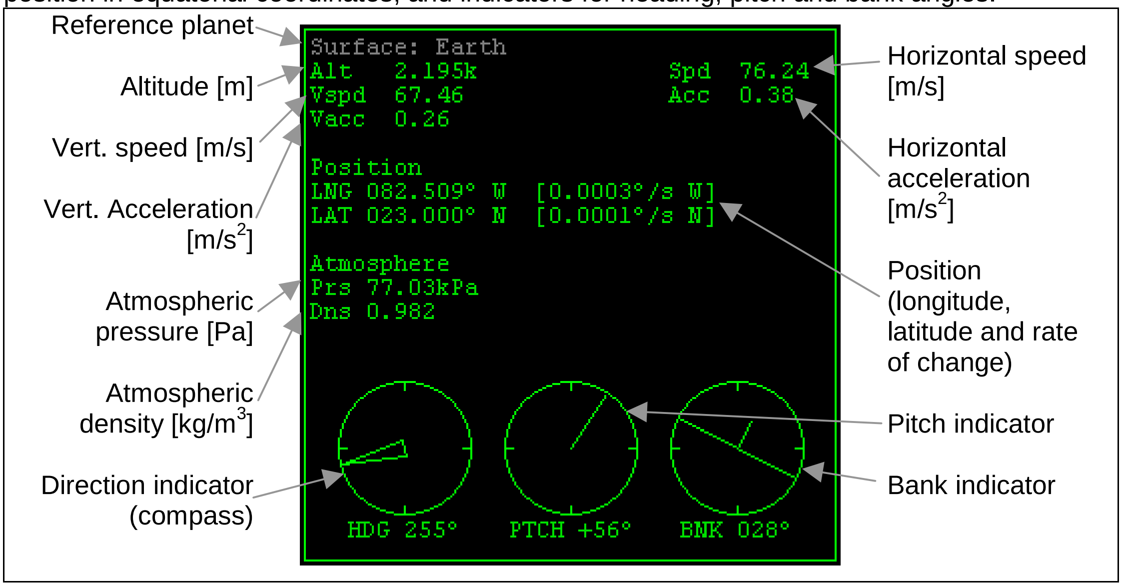


Figure 17: Surface MFD.

11.5. Map

The *Map* MFD mode shows a surface map of a planet or moon, including all available surface bases. The position of the currently selected base and the distance to it are displayed. In addition, the projections of the orbital planes of the ship and the selected orbital target onto the planet surface are plotted.

Key options:

Shift R	Open input box for reference planet/moon selection.
Shift T	Open a menu for target selection.

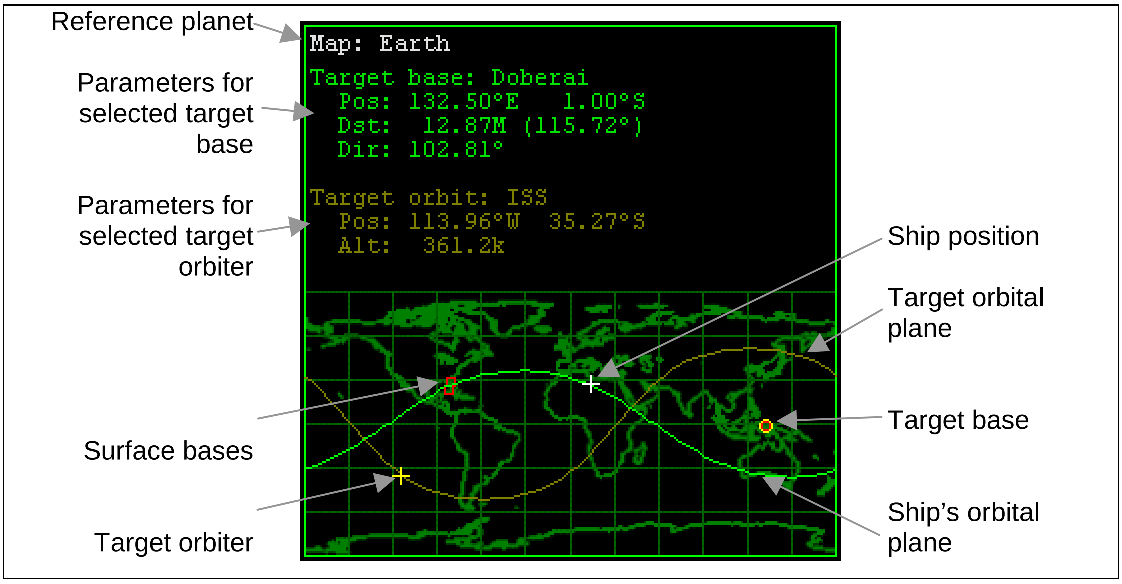


Figure 18: Map MFD.

Target base:

- **Pos**: Equatorial coordinates (longitude, latitude) of selected spaceport.
- **Dst**: Surface distance of ship's projected position to the target base.
- **Dir**: Direction of the target base as seen from the ship.

Target orbit:

- **Pos**: Equatorial coordinates (longitude, latitude) of the projected target position.
- **Alt**: Target altitude.

<u>Notes:</u>

- Only objects (ships, stations or moons) orbiting the current reference planet will be accepted as orbit targets.
- Only bases located on the current reference planet will be accepted as target bases.
- Your ship's orbital plane will only be plotted if you are orbiting the current reference planet.

11.6. Align Orbital Plane

This MFD mode aids in rotating the orbital plane in space so that it corresponds with some target plane, e.g. the orbital plane of another object. The instrument contains the relevant orbital elements (inclination and longitude of the ascending node) of the current and target orbits. It also shows the relative inclination (angle between the two planes), the angles of the current radius vector towards ascending and descending nodes, the time to intercept the next node, and the predicted required thruster burn time. See section 14.3 on how to use this MFD mode.

Key options:

Shift T	Input a new target object or target orbital parameters.

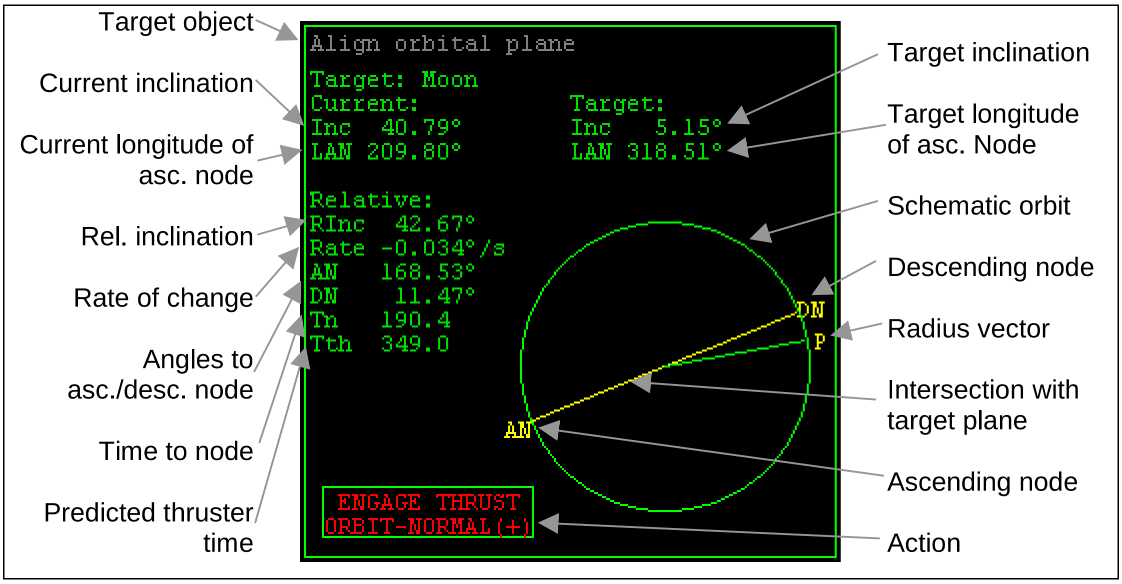


Figure 19: Align Orbital Plane MFD mode.

11.7. Synchronise Orbit

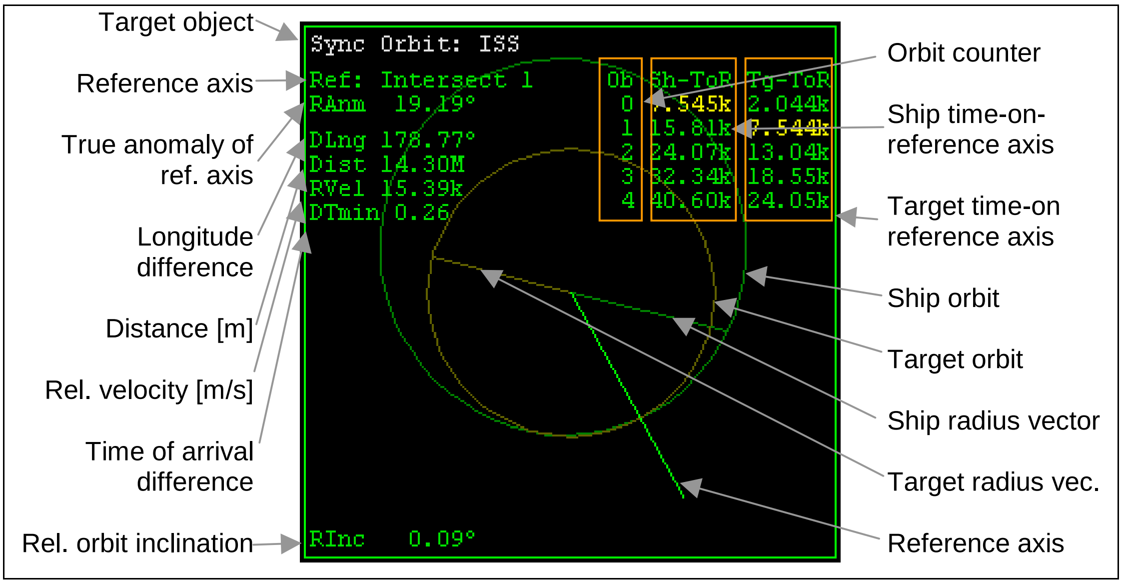


Figure 20: Synchronise Orbit MFD mode.

The Synchronise Orbit MFD assists in catching up with an orbiting body once the orbital planes have been aligned (see previous section).
The instrument displays the ship's and target body's orbits, together with a reference axis and lists the times it will take both objects to reach this axis for a series of orbits.

For this instrument to work properly the orbital planes of both objects must coincide. The relative inclination of the orbital planes is shown in the lower left corner ("RInc"). If this becomes greater than 1° , realign the planes using the*Align Orbital Planes* MFD. Once the planes are aligned, all subsequent manoeuvres should be performed in this plane.

- **Target object**: The synchronisation target is displayed in the title line. It can be selected with Shift T.
- **Reference axis**: A selectable axis for which timings are computed. Can be selected with Shift R from one of the following: orbit intersection 1 and 2 (if applicable), ship and target apoapsis and periapsis, and manual. The manual axis can be rotated with Shift , and Shift ..
- **True anomaly of ref. axis (RAnm)**: The direction of the reference axis w.r.t. the ship's periapsis direction.
- **Longitude difference (DLng)**: The angle between ship and target as seen from the central body.
- **Distance (Dist)**: Distance between ship and target [m].
- **Rel. velocity (RVel)**: Relative velocity between ship and target [m/s].
- **Time-of-arrival difference (DTmin)**: This is the minimum time difference [s] between the ship's and target's arrival at the reference point for any of the listed orbits (see below).
- **Rel. orbit inclination (RInc)**: Inclination between ship's and target's orbital planes.
- **Time-on-reference lists (Sh-ToR and Tg-ToR)**: A list of time intervals for the ship and target to reach the selected reference point. The number of orbits can be selected with Shift N. The closest matched pair of timings is indicated in yellow. The DTmin value refers to this pair.

Key options:

Shift T	Select target object. Only objects orbiting the same body as the ship will be accepted.
Shift R	Select reference axis mode. Intersection 1 and 2 are only available if the orbits intersect.
Shift , / .	Rotate reference axis (manual axis mode only).
Shift N	Select number of orbit timings in the list.

For usage of this MFD mode in orbit synchronisation, see Section 14.5.

11.8. Transfer

The Transfer MFD mode is used for calculating transfer orbits between planets or moons (or more generally, between any objects with significantly different orbits, for which the Sync orbit MFD is not sufficient).

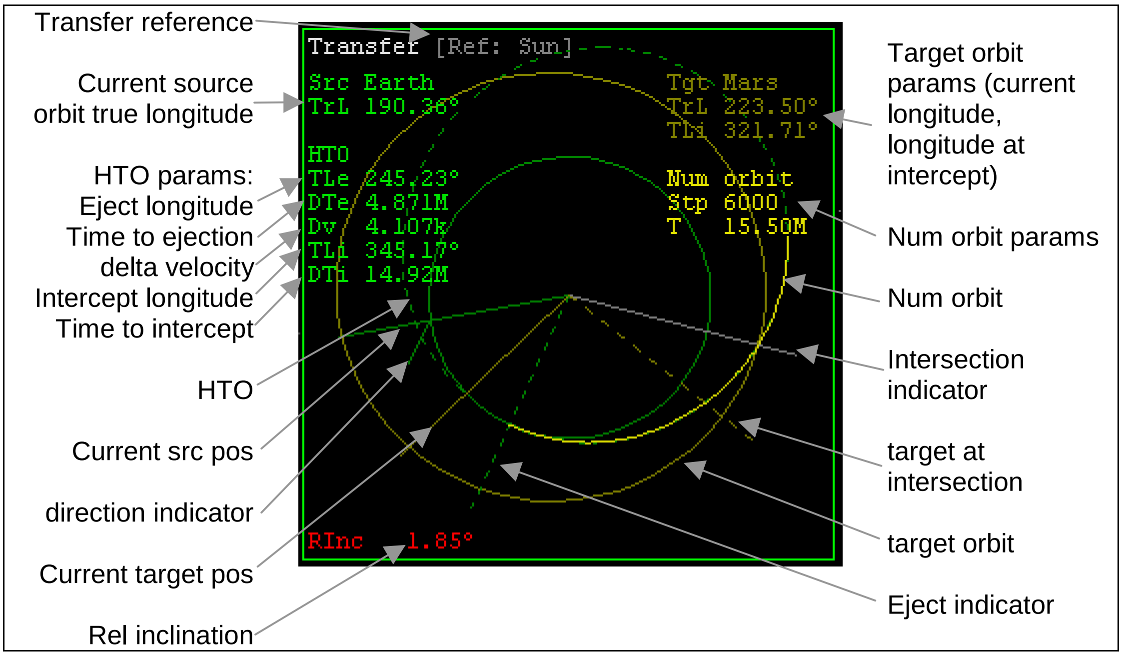


Figure 21: Transfer MFD mode.

The Transfer MFD looks similar to the Orbit MFD: it displays a *source* and a *target* orbit, relative to a selectable orbit *reference*. The source orbit is usually your ship's current orbit, although sometimes a different source is more appropriate (see below). The MFD again assumes matching orbital planes of source and target, although this condition usually can not be precisely satisfied for interplanetary orbits.

Key options:

Shift R	Open input box for reference planet/moon selection.
Shift S	Open a menu for source orbit object selection.
Shift T	Open a menu for target selection.
Shift N	Unselect target.
Shift X	Toggle (hypothetical) transfer orbit display on/off.
Shift M	Toggle numerical multibody trajectory calculation.
Shift U	Refresh numerical trajectory, if displayed.
Shift Z	Open input box for time step definition.
Shift , / .	Rotate transfer orbit ejection longitude.
Shift - / =	Decrease/increase transfer orbit major axis.

Source orbit selection

The source orbit is the orbit from with to eject into the transfer orbit. Usually the source orbit will be the ship's current orbit. In certain situations however it is better to use a different source. Consider for example an interplanetary transfer from Earth to Mars, using the Sun as reference. Since the ship's primary gravitational source will be Earth rather than the Sun, its orbit w.r.t. the Sun will be strongly distorted by the Earth's field. In this case it is better to directly use Earth as the source orbit.

Whenever the source is not identical to the ship, a small direction indicator will be displayed at the current source position which shows the ship's direction w.r.t. the source. This helps with timing the ejection burn (e.g. direction indicator pointing away from the Sun

Hypothetical transfer orbit
Unlike in Orbit mode, this MFD allows you to plot a hypothetical transfer orbit (HTO), which allows to set up “what if” scenarios, without having to change the actual orbit. The HTO display is toggled on/off via [Shift][X]. It is calculated assuming that somewhere along the current source orbit a prograde or retrograde orbit ejection burn occurs. The HTO has two parameters: the *longitude* at which the ejection burn occurs (adjusted with [Shift][,]/[.]) and the velocity change during the burn (adjusted with [Shift][-]/[=]). The HTO is displayed as a dashed green curve in the MFD. The position of the ejection burn is indicated by a dashed green radius vector.

A number of parameters is shown when the HTO is turned on:
TLe: True longitude of orbit ejection point
DTe: Time to ejection point [s]
Dv: Velocity difference resulting from ejection burn [m/s]
TLi: True longitude of interception with target orbit (if applicable)
DTi: Time to interception with target orbit [s] (if applicable)

Intercept indicator
If the source orbit (or, if shown, the HTO) intersects the target orbit, the intersection point is marked by a grey line, and the intersection longitude is displayed (TLi). The position of the target at the time when the ship reaches the intersection point is marked by a dashed yellow line. *The objective is to adjust the HTO so that the grey and dashed yellow lines coincide, so that ship and target arrive at the intersection point simultaneously.*

Hohmann transfer orbit
A transfer orbit which just touches the target orbit (i.e. where ejection and intersection longitude are 180° apart) is called a Hohmann minimum energy transfer orbit, because it minimises the amount of fuel used during the orbit ejection and injection points. Transfer orbits with larger major axis require more fuel, but are faster than Hohmann orbits.

Ejection burn
Once the HTO has been set up, the ejection burn takes place when the ejection point is reached (when the solid and dashed green lines coincide). The ejection burn is prograde (or retrograde) given the orbit w.r.t. the current orbit reference. As the burn takes place, the current orbit (solid green line) will approach the HTO. The burn is terminated when the orbit coincides with the HTO, and Dv has reached zero. After ejection the HTO should be turned off so that intercept parameters are displayed for the actual transfer orbit.

Numerical multibody trajectory calculation
The source, target and transfer orbits discussed above are analytic 2-body solutions. The Transfer MFD however also supports a numerical trajectory calculation, to account for the effect of multiple gravitational sources. The display of the numerical trajectory is toggled with [Shift][M]. The trajectory is displayed as a solid bright yellow line. The calculation is performed in discrete time steps, starting from the current source position, or (if displayed) from the HTO ejection point. Since the calculation of the trajectory can be time-consuming, it is not automatically updated, but can be refreshed with [Shift][U]. The interval between time steps is automatically adjusted to provide consistent accuracy. The number of time steps, and thus the length of the trajectory, can be selected via [Shift][Z]. The number of time steps, and the total time interval covered by the trajectory, are displayed under “Num orbit” in the MFD.

Interplanetary transfers
Using the Transfer MFD for Earth to Moon orbits should be straightforward. For interplanetary transfers (e.g. Earth to Mars) a few caveats apply:

- For interplanetary transfers, the reference should be the Sun, and the source orbit should be the *planet currently being orbited*. This is because the ship’s orbit w.r.t. the Sun will be severely distorted by the planet.
- The ship should be in an orbit with zero inclination against the ecliptic before ejection. The relative inclination between source and target orbits cannot be adjusted, it is simply given by the relative inclination between the planets’ orbits.

- The ejection burn should take place with the Sun in opposition (on the planet's "dark" side) so that the ship's orbital velocity is added to the planetary velocity. This is the case when the source→ship direction indicator is pointing away from the Sun.
- Immediately before the ejection burn, switch the source orbit to your ship, so that Dv can be estimated.

11.9. Ascent profile (custom MFD mode)

This MFD mode is only available if the "Custom MFD" plugin is activated in the *Modules* section of the Launchpad dialog. The Ascent profile mode can be selected with Shift F1-Shift P. The ascent profile records a number of spacecraft parameters and displays them in graphs on the MFD. The following are recorded:

- Altitude as a function of time.
- Pitch angle as a function of altitude.
- Radial velocity as a function of altitude.
- Tangential velocity as a function of altitude.

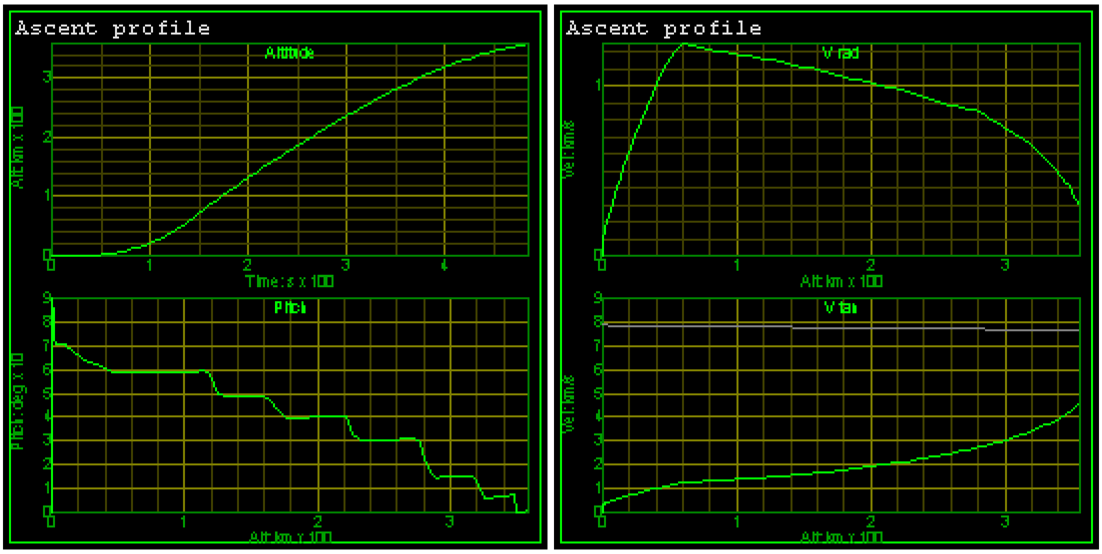


Figure 22: Ascent profile MFD mode, pages 1 and 2

Key options:

Shift P	Switch display page.
Shift A	Set altitude range.
Shift R	Set radial velocity range.
Shift T	Set tangential velocity range.

Parameters are sampled at 5-second intervals. A total of 200 samples are stored and cycled. By default, axis ranges are adjusted automatically, but manual range setting is possible.

Circular orbit insertion

In the tangential velocity graph (Vtan), a grey line indicates the orbital velocity for a circular orbit as a function of altitude. If the vessel's tangential velocity crosses this line for a given altitude, while simultaneously the radial velocity crosses zero, circular orbit is achieved.

12. Quickstart

At the start of the program you are inside a spacecraft located on a launch pad at Habana International Spaceport.

Have a look at your ship:

- Press F1 to switch to an external camera. Use Ctrl+Arrow keys to move the camera around your craft.
- Pressing F1 again from any external camera mode gets you back inside the cockpit.

What else is there?
- Press Ctrl+F1 to bring up a list of objects in the solar system. Use the up and down arrow keys to highlight an object, and Enter to select it.
- Objects with an '>' indicator have associated objects which can be selected with the right arrow key. To return from a sub-menu press the left arrow key.
- Use Ctrl+Arrow keys to move the camera around the selected object. PgUp and PgDown moves the camera in or out. The 'z' and 'x' keys zoom in and out.

Head up display
- Press F1 to get back into the cockpit.
- Press Ctrl+H to activate the HUD.
- Press H until Surface mode is selected ('SRFCE' in the upper left corner of the screen)
- Press LeftShift+S to bring up the left MFD in Surface mode.
- Press RightShift+L to bring up the right MFD in Landing mode.

Launch
- Press Ctrl+C to request launch clearance from Space Traffic Control.
- When clearance is given, press 'Ins' on the numerical keypad to engage launch thrusters. Press until your acceleration is $10 m/s^2$.
- Once you have reached about 100m altitude, point the nose upward about 30 degrees (by pulling on your joystick or pressing the Down arrow on the num-keypad) and fire your main engines (using the joystick throttle control or pressing Ctrl+'+' on the num-keypad).
- If your craft has airfoils (like the glider) which produce a lift vector during atmospheric flight, the hover thrusters can slowly be reduced (‘Del’ on the numeric keypad) as airspeed is picking up.

Free flight
- Practise handling your spacecraft, but keep an eye on the altitude indicator!
- Ctrl+'Keypad+' increases main thrusters/reduces retro thrusters.
- Ctrl+'Keypad-' reduces main thrusters/increases retro thrusters.
- Ctrl+'Keypad*' resets main/retro thrusters to zero.
- Ins increases hover thrusters.
- Del reduces hover thrusters.
- '8' and '2' on the num-keypad engage the pitch attitude thrusters.
- '4' and '6' on the num-keypad engage the roll attitude thrusters.
- '1' and '3' on the num-keypad engage the yaw attitude thrusters.
- '5' on the num-keypad engages the 'kill rotation' sequence.
- When you get to high altitudes you will notice how the handling of the ship changes as atmospheric pressure drops.
- The "Hold altitude" (keyboard 'A') and "Keep level with horizon" (keyboard 'L') functions may help stabilise the ship at low altitudes.

Landing
- Press RightShift+M to bring up the Map display.
- Press RightShift+B until Cape Canaveral is selected.
- Press RightShift+L to bring up the Landing display for your landing target.
- Fly towards Cape Canaveral (should roughly be north of your position). Use the indicator on the compass ribbon at the top of the HUD and the target indicator (green cross) on the Landing display for directions.
- Once within 100km of the base press Ctrl+C to request landing clearance.
- Move towards your assigned pad (flashing green). Use the Landing display to align your vessel above the target at altitude ~1km (that's the tricky bit). You can make position adjustments by tilting the ship to get a horizontal acceleration component from the hover thrusters (similar to a helicopter).

- Reduce the hover thruster settings so that you start sinking. Reduce your vertical velocity as you approach the ground (the vertical velocity indicator should not turn red until about 10m above ground).
- You should now be sitting in the middle of the launch pad.

What next?

- Check out Section 14 (Basic Flight Maneuvers) for more information on surface and orbital flights.
- You could try to rendezvous with the ISS (International Space Station) orbiting the Earth. This requires several orbit synchronisation and intercept manoeuvres. Use the *Orbit*, *Align Orbital Plane*, *Synchronise Orbit* and *Docking* MFD modes to assist in the necessary steps.
- You could try to land at the Moon. The transit will take several hours but you can engage time warp mode ('T' to increase, 'R' to reduce). Try to get into a stable orbit around the Moon. There is also a space station you can try to find.

13. Spacecraft Controls

13.1. Main, retro and hover engines

Main thrusters accelerate the ship forward, *retro thrusters* accelerate it backward. Main and retro engines can be adjusted with Ctrl + *Numpad* (to increase main thrust or decrease retro thrust) and Ctrl - *Numpad* (to decrease main thrust or increase retro thrust). Main and retro thrusters can be killed with Ctrl * *Numpad*. The permanent setting can be temporarily overridden with + *Numpad* (set main thrusters to 100%) and - *Numpad* (set retro thrusters to 100%). If available, a joystick throttle control can be used to set main thrusters.

The ship's acceleration $\mathbf{a}$ resulting from engaging main or retro thrusters depends on the force $\mathbf{F}$ produced by the engine and the ship's mass m:

$$\mathbf{F} = m\mathbf{a}$$

In the absence of additional forces (such as gravitation or atmospheric drag) the spacecraft will move with constant velocity $\mathbf{v}$ as long as no engines are engaged. When engines are engaged, the ship's velocity will change according to

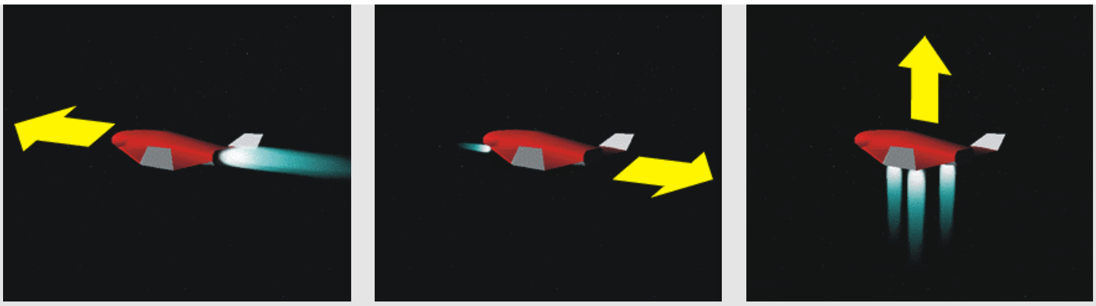
$$\frac{d\mathbf{v}(t)}{dt} = \mathbf{a}(t) \quad \text{or} \quad \mathbf{v}(t) = \mathbf{v}(t_0) + \int_{t_0}^{t} \mathbf{a}(t')dt'$$

The current main/retro thruster setting and corresponding acceleration is displayed in the upper left corner of the HUD ("Main"). The indicator bar is green for positive (main) thrust, and yellow for negative (retro) thrust. The numerical acceleration value is in units of m/s^2. Note that for a fixed thruster setting $\mathbf{F}$ the acceleration will slowly increase as fuel is consumed, resulting in a reduction of the ship's mass m.

Hover engines, if available, are mounted underneath the ship's fuselage to provide upward thrust. Hover thrust is increased with 0 *Numpad* and decreased with . *Numpad*. Hover thrusters are useful to compensate for gravitational forces without the need to tilt the ship upward to obtain an upward acceleration component from the main thrusters.

Spacecraft equipped with airfoils moving within a planetary atmosphere usually do not require hover thrusters except for launch and landing, because they produce an upward force (lift) when moving with sufficient airspeed, like a normal aircraft. Lift is speed-dependent and will collapse below a threshold speed (stall speed).

Main	Retro	Hover



Ctrl + *Numpad* (perm.) + *Numpad* (temp 100%.) Joystick throttle control	Ctrl - *Numpad* (perm.) - *Numpad* (temp 100%.)	0 *Numpad* . *Numpad*

Figure 23: Acceleration from main, retro and hover thrusters.

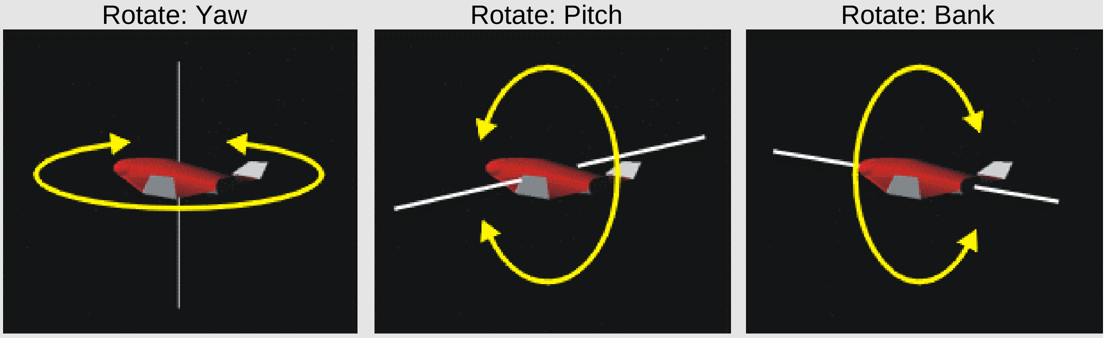
The maximum thrust ratings for main, retro and hover thrusters as well as the current spacecraft mass are displayed in external spacecraft views. Values are in Newton (1N = 1kg m s^{-2}).

13.2. Attitude thrusters

Attitude thrusters are small engines which are engaged in pairs to enable rotation or translation of the spacecraft. In rotation mode, attitude thrusters are fired in cross-linked pairs to produce a rotational moment (e.g. front right and back left to rotate left). In translation mode, thrusters are fired in parallel pairs to produce a linear moment (e.g. front right and back right to accelerate left). The current attitude mode is indicated in the top left corner of the HUD (*Att ROT* and *Att LIN*) and can be toggled with / *Numpad*.

Attitude thrusters are controlled with the joystick or keyboard. In rotation mode:

Rotate: Yaw	Rotate: Pitch	Rotate: Bank

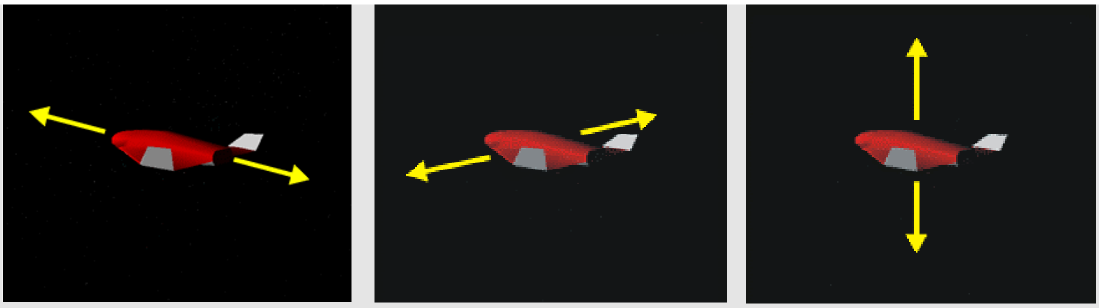


1 *Numpad* 3 *Numpad* Joystick rudder control Or Joystick left/right+Button 2	2 *Numpad* 8 *Numpad* Joystick forward/back	4 *Numpad* 6 *Numpad* Joystick left/right

Figure 24: Attitude thrusters in rotational mode.

In translation mode the spacecraft can be linearly accelerated forward/back, left/right and up/down.

Translate Forward/back	Translate: Left/right	Translate: Up/down



6 Numpad 9 Numpad

1 Numpad 3 Numpad
Joystick rudder control
Or
Joystick left/right+Button 2

2 Numpad 8 Numpad
Joystick forward/back

Figure 25: Attitude thrusters in translational (linear) mode.

For fine control of attitude thrusters with the keyboard use Ctrl-Numpad key combinations. This engages the engines at 10% thrust.
An important control function is the *Kill rotation* sequence (5 *Numpad*). This will automatically engage appropriate attitude thrusters to stop the ship's rotation. The kill rotation sequence only works in rotational attitude mode.

14. Basic Flight Manoeuvres

The following flight techniques are mostly my own invention. They seem plausible, but since I am not a space flight expert (although an enthusiastic amateur) they may be inefficient or plain wrong. Corrections and suggestions are always welcome.

14.1. Surface Flight

By *surface flight* I mean flight paths close to a planetary surface which are not actually orbits, i.e. where the gravitational field of the planet must be countered by applying an acceleration vector, rather than the free fall situation of an orbit. Surface-to-surface transfers (from one surface base to another) typically involve surface flight.

If the planet has no atmosphere

In this case the only forces acting on your ship are the planet's gravitational field and whatever thrust vectors you apply. Most notably, there is no atmospheric friction to reduce the ship's "airspeed". This causes a flight model rather different from a normal airplane. The simplest, but probably not the most efficient strategy for surface flight is:

- Use hover thrusters to balance gravitational acceleration (can be done automatically with "Hold altitude" nav mode). This also means the ship should be kept level with the horizon.
- Navigate with short main thruster bursts.
- At high horizontal velocities the flight path may approach an orbital trajectory. In that case hover thrusters must be reduced to maintain altitude. In the extreme case of horizontal velocity exceeding the orbital velocity of a circular orbit at zero altitude, the ship will gain altitude even for disengaged hover thrusters. That means you have entered into an elliptic orbit at periapsis.

If the planet has an atmosphere

When flying through an atmosphere, the flight model will be similar to an airplane's, in particular if your ship essentially *is* an airplane, i.e. has airfoils that produce a lift vector as a function of airspeed. As with an airplane, you need to apply continuous thrust to counter atmospheric friction and maintain a constant airspeed. If your ship produces lift, hover thrusters are not necessary unless airspeed falls below stall speed (e.g. during vertical lift-off and landing). If your ship does not generate a lift vector, hover thrusters must be substituted, or the ship must be tilted such that the main thrusters provide a vertical component to counter the gravitational field. Note that "lift" produced by thrusters is independent of airspeed.

14.2. Launching into Orbit

Launching from a planetary surface and entering into a low orbit is one of the most basic problems of space flight. During the early part of the launch the ship needs to apply vertical thrust to overcome the gravitational field and acquire altitude. As the ship approaches the desired altitude, the pitch is reduced to increase the horizontal acceleration component, in order to reach orbital velocity. A stable orbit is achieved as soon as the periapsis distance is sufficiently high above the planetary surface so that atmospheric friction can be neglected. Orbits should usually be prograde i.e. rotate in the same direction as the planet surface, to exploit the initial velocity vector provided by the planet. (That is, on Earth ships should be launched eastwards). This also means that launch sites near the equator are most efficient since they provide the largest initial velocity.

In Practice:

(This assumes the ship is initially placed on the Earth's surface).

- Set HUD to surface mode. Bring up Surface (Shift-S) and Orbit (Shift-O) MFD modes.
- Engage hover thrusters to at least $10 m/s^2$.
- Once free of the surface, turn towards east (90° on HUD compass ribbon).
- Raise nose to 70° pitch, while at the same time engaging full main thrusters.
- As air speed increases, bring hover thrusters slowly back to zero.
- As you gain altitude, slowly reduce pitch (e.g. 60° at 20km, 50° at 50km, 40° at 80km, etc.
- As the desired altitude is reached (e.g. 200km) the vertical velocity and acceleration should fall to zero. (by reducing pitch, not by killing the thrusters). Pitch may still be > 0 because part of the thrust vector is required to counter gravitation until full orbital velocity is reached.
- As the tangential velocity increases, pitch should be reduced to maintain constant altitude.
- As soon as the tangential velocity for a circular orbit is reached (eccentricity = 0) thrusters should be killed.

14.3. Changing the Orbit

To change the shape of the orbit without changing the orbital plane, the thrust vector must be applied in the orbital plane. The simplest maneuvers involve modifying the apoapsis or periapsis distances.

- Increase apoapsis distance: Wait until the ship reaches periapsis. Apply thrust vector prograde (ship orientated along velocity vector, engage main thrusters).
- Decrease apoapsis distance: Wait until the ship reaches periapsis. Apply thrust vector retrograde (ship orientated against velocity vector, engage main thrusters).
- Increase periapsis distance: Wait until the ship reaches apoapsis. Apply thrust vector prograde.
- Decrease periapsis distance: Wait until the ship reaches apoapsis. Apply thrust vector retrograde.

In Practice:

Case 1: Assume you want to change from a low circular orbit (200km) into a higher circular orbit (1000km).

- Turn ship prograde and engage main thrusters.
- Kill thrusters as soon as apoapsis distance reaches 1000km+planet radius (e.g. 7370km for Earth). Use *Orbit* MFD mode to monitor this.
- Wait until you reach apoapsis.
- Turn ship prograde and engage main thrusters.
- Kill thrusters as soon as periapsis equals apoapsis and eccentricity is back to 0.

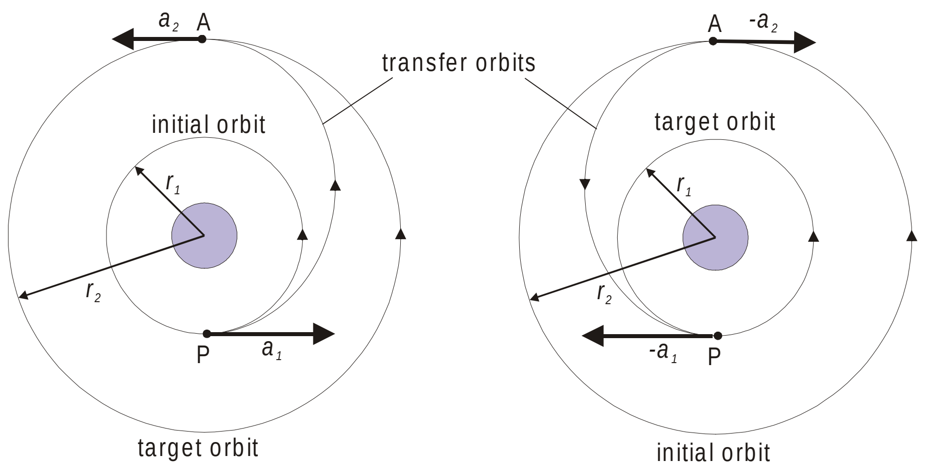


Figure 26: Moving into a higher orbit involves prograde acceleration at P and A (periapsis and apoapsis of the transfer orbit). Conversely, moving from the higher to the lower orbit requires retrograde acceleration at A and P.

Case 2: Rotate the argument of periapsis of an elliptic orbit (i.e. rotate the orbital ellipse in its plane).

- Wait until you reach periapsis.
- Turn ship retrograde and engage main thrusters until orbit is circular (eccentricity = 0).
- Wait until you reach the desired new periapsis position.
- Turn ship prograde and engage main thrusters until original eccentricity and apoapsis distances are re-established.

14.4. Rotating the Orbital Plane

In order to rendezvous with another orbiting body (e.g. a space station) or to prepare for transit to a moon or planet, the first step is to align the orbital plane (OP) of your ship with that of the target. Once you are in the same OP as your target, most of the following navigational problems become essentially two-dimensional, which makes them more robust and a lot easier to compute.

In terms of the orbital elements, aligning the plane of the orbit with a target plane means to match the two elements which define the orientation of the orbit in space: inclination (*i*) and longitude of the ascending node (Ω).

The principal technique to rotate the OP is to point the spacecraft normal (perpendicular) to the current orbital plane, and to fire the engines. This will start to rotate the OP around an axis defined by your current radius vector. Therefore in order to align the orbit with a given target plane:

- Wait until you reach the intersection (node) of your orbit with the target plane
- Rotate the ship to point normal to the current orbit
- Fire the engines until the OP aligns with the target plane.

<u>Note:</u>

- If the angle between the initial and target OP is large it may be necessary to adjust the orientation of the spacecraft during the maneuver to keep it normal to the OP.
- It may not be possible to align the plane in a single node crossing. If the angle towards the target plane cannot be reduced further by accelerating normal to the current orbit, cut the engines and wait for the next node crossing.
- Since the maneuver will take a finite amount of time ΔT, thrusters should be engaged approximately ½ ΔT before intercepting the node.

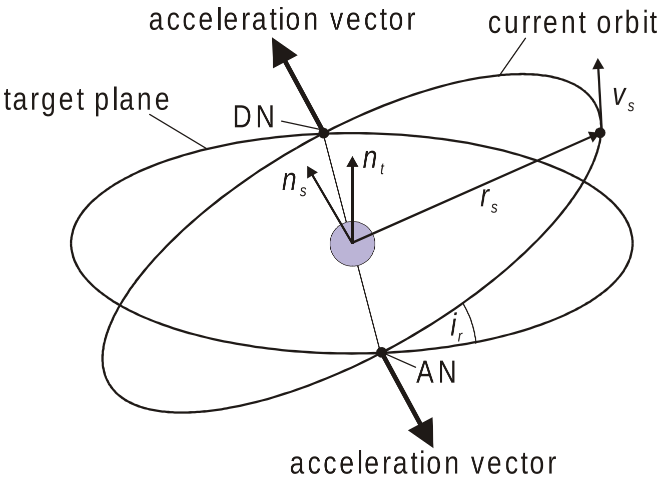


Figure 27: Alignment of the orbital plane. r_s: radius vector. v_s: velocity vector. AN: ascending node. DN: Descending node. n_s: normal of the current plane. n_t: normal of the target plane.

The direction of the normal vector n_s is defined by the direction of the cross product r_s x v_s . Acceleration should be applied in direction $-n_s$ in the ascending node (AN), and in direction $+n_s$ in the descending node(DN). (see Figure 27).

In Practice:

- The *Align orbital plane* MFD mode (Shift-A, see Section 11.6) is designed to aid in plane alignment. Select the target object (Shift-T).
- The HUD should be in Orbit mode. As your ship approaches the intersection with the target plane, rotate it normal (if at DN) or anti-normal (if at AN) to the current orbital plane. Use the HUD Orbit inclination ladder for this.
- As soon as the time to node (Tn) reaches half the estimated burn time (Tth) the “Engage thruster” indicator will start flashing. Engage full main thrusters. Make sure the relative inclination (RInc) decreases, i.e. the rate of change (Rate) is negative, otherwise you may be pointing in the wrong direction.
- Adjust the ship’s orientation as required to keep normal to orbital plane.
- Disengage thrusters as soon as the action indicator turns back to “Kill thruster”.
- If the relative inclination was not sufficiently reduced repeat the procedure at the next node passage.
- During the manoeuvre make sure your orbit does not become unstable. Watch in particular for the eccentricity (use the Orbit MFD to monitor this).

14.5. Synchronising Orbits

This section assumes that the orbital planes of ship and target have been aligned (see previous section).

The next step in a rendezvous manoeuvre after aligning the orbital planes is to modify the orbit in the plane such that it intercepts the target’s orbit and both ship and target arrive simultaneously at the interception point. Use the *Synchronise Orbit* (Shift-Y) MFD to calculate the appropriate orbit.

For simplicity we first assume that the ship and target are in a circular orbit with the same orbital radius (for synchronising the orbital radius see Section 14.3), i.e. both objects have the same orbital elements except for the mean anomaly. The method for intercepting the target is then as follows:

- Switch the reference mode of the *Synchronise Orbit* MFD to “Manual” and rotate the axis to your current position.
- Turn your ship prograde (using Orbit HUD mode) and fire main thrusters.
- The orbit will become elliptic, with increasing apoapsis distance. Periapsis is your current position. Simultaneously the orbit period and the times to reference axis will increase.
- Kill thrusters as soon as one of the Sh-ToR times coincides with one of the Tg-ToR times.
- Then you just have to wait until you intercept the target at the reference axis.

- At interception, fire thrusters retrograde to get back to the circular orbit and match velocity with the target.

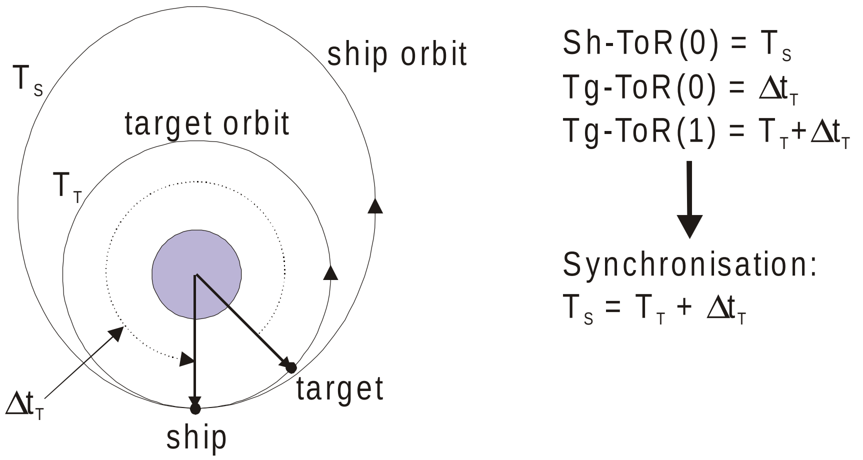


Figure 28: A transition orbit to intercept the target at the next periapsis passage.

<u>Notes:</u>

- Instead of increasing the apoapsis distance one could fire retrograde and reduce the periapsis distance in this manoeuvre. This may be more efficient if the target is ahead of the ship. But make sure that periapsis does not become dangerously low!
- It should always be possible to match your next ToR (orbit 0) with the target's ToR at orbit 1. If you are low on fuel it may however be better to match later orbits if this can be achieved with less distortion to the original orbit. For example, if the target is marginally ahead, then to intercept it in the next orbit you need to nearly double your orbital period.
- It is not essential that the orbits are identical or circular at the start of the manoeuvre. It is sufficient for them to intersect. In that case it is best to use Intersection 1 or 2 reference mode in the *Synchronise* MFD.
- You don't necessarily need to wait until you reach the reference point before firing thrusters, but it simplifies matters because otherwise the intersection point itself will move, making the alignment of orbit timings more difficult.

14.6. Docking

Docking to an orbital station is the last step in the rendezvous manouevre. Assuming you have intercepted the target station following the preceding steps, here we discuss the final docking approach.

- Turn on the *Docking* MFD (Shift-D) and the *Docking* HUD by pressing 'H' until docking mode is selected.
- If not done already, synchronise relative velocity by turning the ship until it is aligned with the relative velocity marker (⊕) and fire main thrusters until velocity value (V) approaches zero.
- Rotate the ship to face the station (marker).
- Request docking clearance (Ctrl-D). Once clearance is given, the HUD will display rectangular approach path indicators, and the alignment indicators in the Docking MFD will be activated.
- Move towards the approach path rectangle furthest away from the station and hold.
- Align the ship's heading with the flight path direction using the 'X' indicator in the MFD.
- Align the ship's position on the approach path using the '+' indicator in the MFD. Switch attitude thrusters to linear mode for this.
- Align the ship's rotation along its longitudinal axis using the arrow indicator in the MFD.
- Approach the station by engaging main thrusters briefly. During approach correct your position continuously using linear attitude thrusters.
- Slow down approach speed to less than 0.1m/s before intercepting the dock.
- You need to approach the dock to less than 0.3 m for a successful docking manoeuvre.

- Once docked, your ship will automatically be refueled. As with spaceports, you can jump into any other ship currently docked to the station by pressing F3.
- To undock, request clearance again with Shift-D.

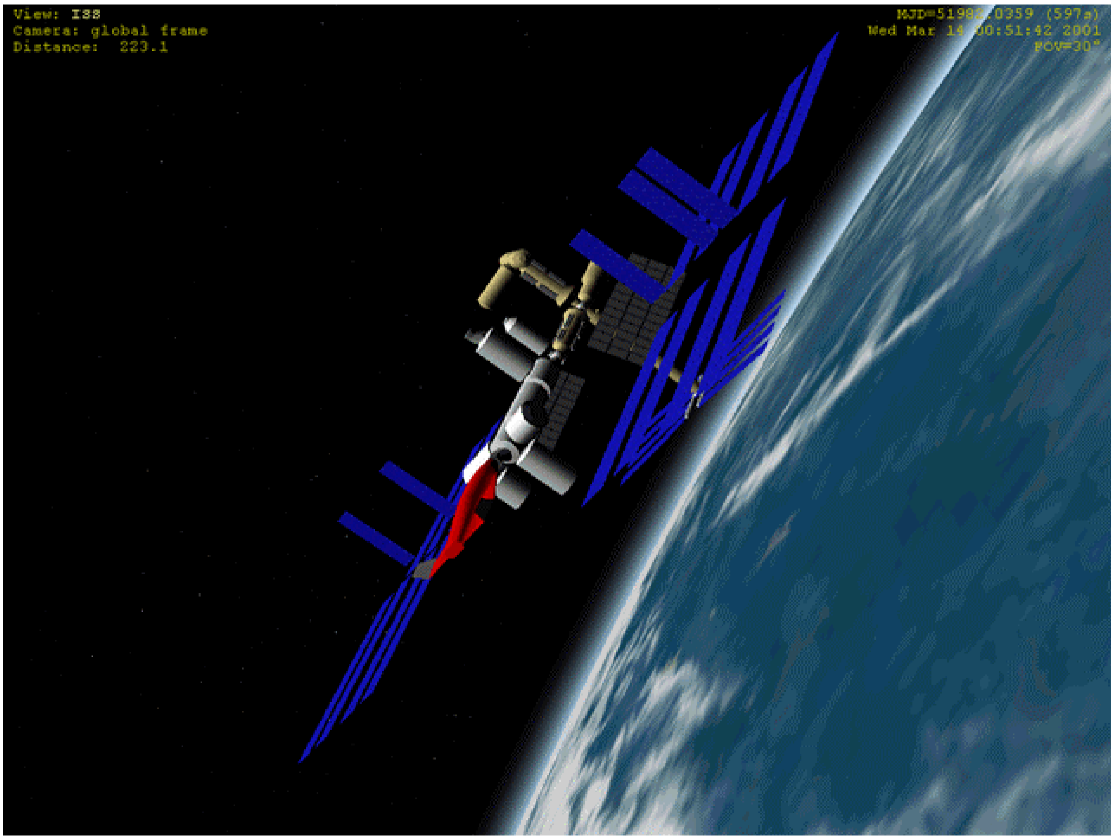


Figure 29: A successfully completed docking approach to the ISS.

<u>Notes:</u>
- For precise attitude control with the keyboard use the attitude thrusters in “low power” mode (Ctrl + Numpad key).
- You can only dock to a port you have been cleared for.
- Rotational alignment is not currently enforced, but may be in future versions.
- Currently no collision tests are performed, so you might fly straight through the station if you miss the docking approach.

Docking at rotating stations

Stations like Luna-OB1 rotate to use centrifugal forces for emulating gravity – which is nice for their inhabitants but makes docking a bit more complicated. Docking is only possible along the rotation axis, so at most 2 docking ports can be provided. The docking procedure is similar to the standard one, but once aligned with the approach path, the rotation around the ship’s longitudinal axis must be aligned with that of the station.

<u>Important:</u>
- Initiate your ship’s longitudinal rotation only immediately before docking (when past the last approach marker). Once you are rotating, linear adjustments become very difficult.
- Once the rotation is matched with the station, don’t hit Numpad5 (Kill rotation) by accident, or you will have to start the rotation alignment again.

<u>Cheat:</u>
Since rotational alignment is not enforced at present, you can simply ignore the rotation of the station and fly straight in.

15. Flight Checklists

This section contains point-by-point checklists for some complete flights.

15.1. Launch from Habana for the International Space Station

This checklist assumes you are currently parked on a Habana Spaceport launch pad. Modify configuration files if necessary.
For this flight, avoid 1000x time acceleration because it will degrade the accuracy of orbit calculation. Remember that spacecraft controls are disabled at time acceleration > 1x.

- Press LShift-M to bring up the Map MFD. Press LShift-S until the ISS is selected as docking target. The yellow sinusoidal curve is the orbital plane of the ISS, projected on the Earth map. The white cross is your current position.
- Wait for your launch window. You should launch as soon the yellow curve intercepts your position. This could take several hours, so you may need to engage time acceleration (T). Switch back to real-time (R) as soon as the ISS orbit intersects your position.
- Press RShift-L to bring up the Launch/Land MFD.
- Press Ctrl-H, H to bring up the HUD in Surface mode.
- Press Ctrl-C to request launch clearance. Wait until you get clearance (shown in the Launch MFD).
- Engage Hover thrusters by pressing Numpad-Ins until you get acceleration >= 10 m/s^2 (‘Hovr’ indicator in upper left corner).
- After takeoff turn towards heading 45° or 135° using attitude thrusters Numpad-1 and Numpad-3. Turn to 45° if you are intercepting the rising part of the ISS orbit, and to 135° if you are intercepting the falling part of the orbit. (The point is to approximately align your orbital plane with the ISS so that only small corrections are required later).
- At altitude > 100 m engage main thrusters fully (Ctrl-Numpad+) and lift the nose to 70° pitch, using Numpad-2 and Numpad-8.
- Kill hover thrusters (Numpad-Del).
- Switch to Surface MFD (LShift-S).
- Switch to Orbit MFD (RShift-O). Select the ship’s orbit as reference plane (RShift-P) and select ISS as target (RShift-T, ‘ISS’).
- Reduce pitch gradually as you gain altitude: 60° at 20km, 50° at 60km, 40° at 100km, 30° at 150km, 20° at 200km, 10° at 270km, 0° at 340°. Maintain heading during climb. The objective is to reach tangential orbital velocity at ~360km (the altitude of the ISS). During the final stage of the orbit injection you will need to adjust the pitch to drop to zero vertical speed (‘VSpd’ in surface MFD) when reaching altitude 360km. The exact altitude is not critical at this stage.
- Kill thrusters (Numpad*) when the eccentricity of the orbit is close to zero (‘Ecc’ in orbit MFD). You have now entered a circular orbit.
- Switch to Orbit HUD mode (H, H)
- Switch to Align Orbital Plane MFD (LShift-A). Select ISS (LShift-T, ‘ISS’).
- Ideally the orbital planes should already be roughly aligned (RInc within 5°). You now need to fine-adjust the plane.
- As your ship (P) approaches an intersection point with the target plane (AN or DN): Rotate the ship perpendicular to your current orbital plane (90° on the Orbit HUD inclination ladder). If you are approaching the ascending node (AN), turn *south*. If you are approaching the descending node (DN), turn *north* (as explained in Section 14.4).
- As soon as the “Engage engines” indicator starts flashing, engage full main engines. The relative inclination between the orbital planes should now decrease.
- Kill thrusters as soon as the “Kill thrust” indicator appears. If you could not reduce the orbit inclination sufficiently (within 0.5°) repeat the process at the next nodal point.
- Once the planes are aligned, the next step is intercepting the ISS. Switch to Synchronise Orbit MFD (LShift-Y). Select “Manual” reference mode (LShift-R) and rotate the reference axis until it points just behind your current position. The time-on-reference indicator for orbit 0 (Sh-ToR(0)) should now approximately equal the orbital period.
- Turn the ship prograde (align with “⊕” velocity marker). Fire main engines until Sh-ToR(0) matches Tg-ToR(1). You will now intercept the ISS at your next passage of the reference point. You may want to engage time acceleration now until you reach the reference point.

- Switch to Docking HUD mode (H, H) and to Docking MFD (LShift-D).
- Rotate the ship to align with the relative velocity marker ("⊕") and fire main engines until relative velocity is close to zero.
- Rotate the ship towards the ISS (target designator box) and move to within 5km of the station. You may want to use attitude thrusters in linear (translatorial) mode for this. Switch between linear and rotational mode with the Numpad/ key.
- Request docking clearance (Ctrl-D).
- When cleared for docking, the approach path indicator in the Docking MFD will become active, and the approach path will be shown as a series of rectangles on the HUD.
- Move towards the rectangle furthest away from the station and hold.
- Align your ship's longitudinal axis with the approach path direction (align "X" indicator in the MFD) using attitude thrusters in rotational mode.
- Align your ship's rotation around its longitudinal axis (align "^" indicator at 12 o'clock position in the MFD).
- Center your ship on the approach path (align "+" indicator in the MFD) using linear attitude thrusters.
- Start moving towards the dock with a short burst of the main engines. Closing speed (CVel) should be gradually reduced as you approach the dock. Final speed should be < 0.1 m/s. Re-align ship on the approach path with linear attitude thrusters as required.
- The docking mechanism should engage once you are within 0.3 m of the designated dock. A "Dock" indicator will appear in the MFD once your ship has successfully docked.
- Finished!

16. ORBITER Configuration

Configuration files allow the customisation of various aspects of ORBITER. Configuration files have file extension .cfg. The format is

item = value

A semicolon starts a comment, continuing to the end of line.
All configuration files except for the master file (see below) are located in a subdirectory defined by the ConfigDir entry in the master file, usually ".\Config".
Scenarios (simulation startup definitions) are located in a subdirectory defined by the ScenarioDir entry in the master file, usually ".\Scenarios". They have file extension .scn.

16.1. Master Configuration File

The master configuration file *Orbiter.cfg* is located in the Orbiter main directory. It contains general settings for graphics modes, subdirectory locations, simulation parameters, etc. Note that manual editing of Orbiter.cfg should no longer be necessary, because most parameters can be accessed from the Launchpad dialog.

Item	Type	Description
ConfigDir	String	Subdirectory for configuration files (default .\Config\)
MeshDir	String	Subdirectory for mesh files (default .\Meshes\)
TextureDir	String	Subdirectory for textures (default .\Textures\)
HightexDir	String	Subdirectory for alternative high-resolution planetary textures (default: .\Textures2\)
ScenarioDir	String	Subdirectory for scenarios (default: .\Scenarios\)
DeviceIndex	Int	Enumeration index for current 3D device *(do not edit manually)*
ModeIndex	Int	Screen mode index *(do not edit manually)*
Fullscreen	Bool	TRUE for fullscreen mode, FALSE for windowed mode
Stereo	Bool	*Currently not used.*
WindowWidth	Int	Horizontal window size for windowed modes.
WindowHeight	Int	Vertical window size for windowed modes. The ratio WindowWidth/WindowHeight should be approximately 4/3.
JoystickIndex	Int	Enumeration index for current joystick (0=none; *do not edit manually)*
JoystickThrottle	Int	Saturation zone for joystick throttle control (0-10000). A

Saturation		setting of 9000 means that the throttle will saturate over the last 10% of its range at either end. Default: 9500
JoystickDeadzone		Deadzone at joystick axis centres (0-10000). A setting of 2000 means the joystick is considered neutral within 20% from the central position. Default: 2500
AmbientLevel	Int	Ambient light level (brightness of not directly lit surfaces). Valid range is 0-15.
NumStar	Int	Number of background stars (<= 15984). Default: 3000
StarBrightness	Float	Brightness scaling factor for background stars (0.2 …2). Default: 1.0
ConstellationCol	RGB	Colour for constellation lines. Default: 0.3 0.3 0.3
MoveCamPaused	Bool	Enable camera controls during pause. This keeps the render engine going and therefore does not reduce CPU load when the simulation is paused.
UnlimitedFuel	Bool	Ignore spacecraft fuel consumption. Default: FALSE
MFDTransparent	Bool	Make onscreen multifunctional displays transparent. Default: TRUE
EnableShadows	Bool	Enable/disable object shadows on planet surfaces.
InstrumentUpdateInterval	Float	Interval between MFD display updates (seconds)
PlanetPatchRes	Float	Resolution factor for planet surfaces. Valid range is 0.1 to 10. Higher values produce higher resolution planetary surfaces at a given apparent radius, but reduce performance.
DialogFont_Scale	Float	Scaling factor for dialog font size. Default: 1.0
DialogFont1_Face	String	Standard dialog font face. Default: Arial

16.2. Planetary Systems

Planetary systems contain stars, planets and moons. Each planetary system requires at least one star.

Star entries:

Star<i> = Name

where <i> is an index running from 1 upward. *(note: planetary systems with more than one central star are not currently supported).*

Planet entries:

Planet<i> = Name

where <i> is an index running from 1 upward.

Moon entries:

<Planet>:Moon<i> = Name

where <Planet> is the name of a planet defined before, and <i> is an index enumerating the moons of this planet, running from 1 upward.

Example:

Star1 = Sun
Planet1 = Mercury
Planet2 = Venus
Planet3 = Earth
Earth:Moon1 = Moon
Planet4 = Mars
Mars:Moon1 = Phobos
Mars:Moon2 = Deimos

16.3. Planets

Planet configuration files define the planet's orbital, physical and visual parameters. For an example see Config\Earth.cfg.

1. General parameters

Item	Type	Description
Name	String	Planet name
Module	String	Name of dynamic link library performing calculations for the planet (default: none)
ErrorLimit	Float	Max. rel. error for position/velocity calculations (only used if the module supports precision adjustment)
EllipticOrbit	Bool	If TRUE, use analytic 2-body solution for planet position/velocity calculation, otherwise update dynamically (ignored if module supports position/velocity calculation)
HasElements	Bool	If TRUE, the initial position/velocity is calculated from the provided set of orbital elements, otherwise from an explicit position/velocity pair (ignored if module supports position/velocity calculation)

Notes:

- If the module calculates the planet position and velocity from perturbation terms, then the value of ErrorLimit will affect the number of terms used for the calculation. A lower value will increase the number of required terms, and thus the calculation time. The valid range for ErrorLimit depends on the module, but is typically 1e-3 ≤ ErrorLimit ≤ 1e-8.

2. Orbital parameters (Ignored if module supports position/velocity calculation or HasElements = FALSE)

Item	Type	Description
Epoch	Float	Orbital element reference epoch (e.g. 2000)
ElReference	Flag	`ParentEquator` or `Ecliptic`: orbit reference frame (default: Ecliptic)
SemiMajorAxis	Float	Orbit semi-major axis [m]
Eccentricity	Float	Orbit eccentricity
Inclination	Float	Orbit inclination against reference plane [rad]
LongAscNode	Float	Longitude of ascending node [rad]
LongPerihelion	Float	Longitude of periapsis [rad]
MeanLongitude	Float	Mean longitude at epoch [rad]

3. Physical parameters

Item	Type	Description
Mass	Float	Planet mass [kg]
Size	Float	Mean planet radius [m]

4. Rotation elements

Item	Type	Description
SidRotPeriod	Float	Siderial rotation period [s]
SidRotOffset	Float	Rotation at epoch [rad]
Obliquity	Float	Obliquity: angle between rotation axis and normal of reference plane at epoch [rad]
LAN	Float	Longitude of projection of rotation axis onto reference plane [rad]

5. Atmospheric parameters (only required if planet has atmosphere)

Item	Type	Description
AtmPressure0	Float	(Mean) atmospheric pressure at zero altitude [Pa]
AtmDensity0	Float	(Mean) atmospheric density at zero altitude [kg/m^3]
AtmColor0	Vec$_3$	RGB triplet for atmospheric colour at ground level (0-1 each)

AtmAltLimit	Float	altitude limit beyond which atmospheric effects can be ignored [m]

6. Cloud parameters (only required if planet contains a cloud layer)

Item	Type	Description
CloudAlt	Float	Altitude of cloud layer [m]
CloudRotPeriod	Float	Rotation period of cloud layer against surface [s] (default: 0 – static cloud layer)

7. Visualisation parameters

Item	Type	Description
MaxPatchResolution	Int	Max. resolution level for surface texture maps (1 ... 8)
MinCloudResolution	Int	Min. resolution at which clouds are rendered as separate layer (1 ... MaxPatchResolution)
MaxCloudResolution	Int	Max. cloud resolution level (MinCloudResolution ... 8)
MaxDynamicResolution	Int	Reserved

8. Surface bases (only required if planet contains surface bases)

Item	Type	Description
NumBases	Int	Number of spaceports on planet surface
Base<X>	List	<X> = 1 ... NumBases. List contains: Name, longitude [deg], latitude [deg]

To implement a custom DLL module for planet position/velocity calculations, see SDK documentation.

To add a new planet to a planetary system the following steps are required:

1. Add an entry for the planet in the planetary system configuration file (see previous section):
   Planet<X> = <Planetname>

2. Create a configuration file *<Planetname>.cfg* for the new planet in the “Config” subdirectory, with entries as listed above.

3. Create the required surface texture maps up to the specified resolution (see Section 17).

4. Optionally, create a monochrome (green on black) surface outline bitmap (256x128, BMP) to be used by the Map MFD. The file name should be *<Planetname>M.bmp*.

16.4. Station classes

Configuration files for classes of orbital stations. The class definition contains parameters (such as mass, size and mesh) for a *type* of station, but not the parameters (such as position) for *individual* stations. For an example see Config\ISS.cfg. Individual stations are defined in the *scenario file* (see section 16.9).

Item	Type	Description
BaseClass	String	Parent class. Missing entries are taken from this class. Allows the construction of class hierarchies.
MeshName	String	Name of the mesh used for visualisation
Mass	Float	Station mass [kg]
Size	Float	Mean station radius [m]
ZrotFreq	Float	For spinning stations: rotation frequency [Hz]. Default: 0 (i.e. no rotation). Rotation axis is always the station’s local z-axis. This is a nominal value and can be overridden by individual stations.
<Docklist>	List	List of positions and approach directions for docking ports (see

	below).

This is the syntax for <*Docklist*>:

```
BEGIN_DOCKLIST
   <Dock-spec 0>
   <Dock-spec 1>
   . . .
   <Dock-spec n-1>
END_DOCKLIST
```

where <*Dock-spec i*>:

$<x_i>$ $<y_i>$ $<z_i>$ $<dx_i>$ $<dy_i>$ $<dz_i>$ $<rx_i>$ $<ry_i>$ $<rz_i>$

$<x_i>$ $<y_i>$ $<z_i>$ is the reference position of the docking port in the station's local coordinates.
$<dx_i>$ $<dy_i>$ $<dz_i>$ is the direction in which a ship approaches the docking port in the station's local reference frame.
$<rx_i>$ $<ry_i>$ $<rz_i>$ is a reference direction perpendicular to the approach direction used for aligning an approaching ship's rotation along its longitudinal axis.

16.5. Vessel Classes

This section has been moved to 3DModel.doc in the Orbitersdk\doc subdirectory, which is contained in the SDK package.

16.6. Vessels

Note that the definition of individual vessels has changed from previous versions of Orbiter. A vessel only requires a separate definition file if it is not an instance of a vessel class. In this case the format for the vessel .cfg file is identical to the vessel class cfg file.
The time-dependent vessel parameters (state, position, velocity) are now defined in the *Scenario* (.scn) files.

16.7. Surface Bases

To create a new spaceport on the surface of a planet the following steps are required:

1. In the planet's configuration file
   - Increase the entry for *NumBases* by 1, or add the line
     NumBases = 1
     if no bases are defined for the planet yet.
   - Add a line
     Base<X> = <Name> <Long> <Lat>
     where <*X*> is the base's numeric ID (should be set to the new value of *NumBases*), <*Name*> is the base's name (which identifies its configuration file), and <*Long*> and <*Lat*> define the base's position on the planet surface in equatorial coordinates (degrees). Eastern Longitudes and northern Latitudes are counted positive, western longitudes and southern latitudes are negative.

     Example:
     NumBases = 1
     Base1 = Habana -82.5 +23.0

2. Create a configuration file for the base, <*Name*>.cfg, where <*Name*> corresponds to the base's name in the planet configuration file. The format of the base definition file is as follows:

```
NAME = <Base name>
SIZE = <size>

BEGIN_OBJECTLIST
   <Object list>
END_OBJECTLIST
```

where <*Base name*> is the base's (logical) name which need not correspond to the entry in the planet configuration file. <*Size*> is the base's overall radius in meters.

<*Object list*> contains a list of objects which make up the visual elements of the base. See next section for details.

16.8. Adding objects to surface bases

Surface bases are composed of objects (buildings, train lines, hangars, launch pads, etc.) The configuration file for each surface base contains a list of its objects:

```
BEGIN_OBJECTLIST
   <Object 0>
   <Object 1>
   …
   <Object n-1>
END_OBJECTLIST
```

Each object entry in the list defines a particular object and its properties (type, position, size, textures, etc.). An object can either be a pre-defined type or a generic mesh. Each object entry has the following format:

```
<Type>
   <Parameters>
END
```

Note that textures used by base objects must be listed in the texture list of the *Base.cfg* configuration file.

The following pre-defined object types are currently supported:

BLOCK

A 5-sided "brick" (without a floor) which can be used as a simple generic building, or as part of a more complex structure. The following parameters are supported:

Parameter	Type	Description
POS	V	Centre of the block's base rectangle (in local coordinates of the surface base). Note that the y-coordinate is the elevation above ground. Default: 0 0 0
SCALE	V	Object size in the three coordinate axes. Default: 1 1 1
ROT	F	Rotation around vertical axis (degrees). Default: 0
TEX1	S F F	Texture name and u,v scaling factors for walls along the x-axis. Default: none
TEX2	S F F	Texture name and u,v scaling factors for walls along the z-axis. Default: none
TEX3	S F F	Texture name and u,v scaling factors for roof. Default: none

(V=Vector, F=Float, S=String)

HANGAR

A hangar-type building with a barrel-shaped roof. The following parameters are supported:

Parameter	Type	Description
POS	V	Centre of the object's base rectangle (in local coordinates of the surface base). Note that the y-coordinate is the elevation above ground. Default: 0 0 0
SCALE	V	Object size in the three coordinate axes. Default: 1 1 1
ROT	F	Rotation around vertical axis (degrees). Default: 0
TEX1	S F F	Texture name and u,v scaling factors for walls. Default: none
TEX2	S F F	Texture name and u,v scaling factors for front gate. Default: none

TEX3	S F F	Texture name and u,v scaling factors for roof. Default: none

(V=Vector, F=Float, S=String)

HANGAR2

A hangar-type building with a tent-shaped roof. The following parameters are supported:

Parameter	Type	Description
POS	V	Centre of the object's base rectangle (in local coordinates of the surface base). Note that the y-coordinate is the elevation above ground. Default: 0 0 0
SCALE	V	Object size in the three coordinate axes. Default: 1 1 1
ROT	F	Rotation around vertical axis (degrees). Default: 0
TEX1	S F F	Texture name and u,v scaling factors for front and back walls. Default: none
TEX2	S F F	Texture name and u,v scaling factors for side walls. Default: none
TEX3	S F F	Texture name and u,v scaling factors for roof. Default: none
ROOFH	F	Roof height from base to ridge. Default: ½ building height.

(V=Vector, F=Float, S=String)

TANK

A fuel tank-like upright cylinder with flat top. The following parameters are supported:

Parameter	Type	Description
POS	V	Centre of the object's base circle (in local coordinates of the surface base). Note that the y-coordinate is the elevation above ground. Default: 0 0 0
SCALE	V	Cylinder radii in x and z, and height in y. Default: 1 1 1
ROT	F	Rotation around vertical axis (degrees). Default: 0
NSTEP	I	Number of segments to approximate circle. Default: 12
TEX1	S F F	Texture name and u,v scaling factors for mantle. Default: none
TEX2	S F F	Texture name and u,v scaling factors for top.

(V=Vector, F=Float, I=Integer, S=String)

SOLARPLANT

A grid of ground-mounted solar panels, smart enough to align themselves with the Sun. The following parameters are supported:

Parameter	Type	Description
POS	V	Centre position of the panel grid. Default: 0 0 0
SCALE	F	Scaling factor for each panel. Default: 1
SPACING	F F	Distance between panels in x and z direction. Default: 40 40
GRID	I I	Grid dimensions in x and z direction. Default: 2 2
ROT	F	Rotation of plant around vertical axis (degrees). Default: 0
TEX	S [F F]	Texture name and u,v scaling factors for panels. Default: none

(V=Vector, F=Float, I=Integer, S=String)

TRAIN1

A monorail-type train on a straight track. The following parameters are supported:

Parameter	Type	Description
END1	V	First end point of track
END2	V	Second end point of track
MAXSPEED	F	Maximum speed of train [m/s] Default: 30
SLOWZONE	F	Distance over which train slows down at end of track [m] Default: 100
TEX	S	Texture name

(V=Vector, F=Float, S=String)

TRAIN2

Suspension train on a straight track. The following parameters are supported:

Parameter	Type	Description
END1	V	First end point of track
END2	V	Second end point of track
HEIGHT	F	Height of suspension track over ground [m] Default: 11
MAXSPEED	F	Maximum speed of train [m/s] Default: 30
SLOWZONE	F	Distance over which train slows down at end of track [m] Default: 100
TEX	S	Texture name

(V=Vector, F=Float, S=String)

LPAD1

An octagonal bordered landing pad. Default diameter 80m (at scale 1). Landing pads are numbered in the order they appear in the list. Can be assigned numbers 1-9. For expected layout of texture map see e.g. Textures\Lpad01.dds.

Parameter	Type	Description
POS	V	Pad centre coordinates (in local coordinates of the surface base).
SCALE	F	Scaling factor. Default: 1
ROT	F	Rotation around vertical axis (degrees). Default: 0
TEX	S	Texture name. Default: none

(V=Vector, F=Float, S=String)

LPAD2

A square landing pad. Default size 80m (at scale 1). Landing pads are numbered in the order they appear in the list. Can be assigned numbers 1-99. For expected layout of texture map see e.g. Textures\Lpad02.dds.

Parameter	Type	Description
POS	V	Pad centre coordinates (in local coordinates of the surface base).
SCALE	F	Scaling factor. Default: 1
ROT	F	Rotation around vertical axis (degrees). Default: 0
TEX	S	Texture name. Default: none

(V=Vector, F=Float, S=String)

MESH

Generic mesh for custom object types. Mesh files must be in ORBITER mesh file format (see Appendix A). The group texture and material properties defined in the mesh file are ignored.

Parameter	Type	Description
FILE	S	Mesh file name (without path and extension). Mesh files must be located in the mesh subdirectory (see master config file).
POS	V	Position of mesh origin (in local coordinates of the surface base).
SCALE	V	Scaling factors in x and z, and height in y. Default: 1 1 1
ROT	F	Rotation around vertical axis (degrees). Default: 0
TEX	S	Texture name. Default: none
UNDERSHADOWS		Object can be covered by shadows cast on the ground by other objects (e.g. roads, landing pads, etc.). Default: object not covered by ground shadows
LPAD		Object is a landing pad.
PRELOAD		Mesh should be loaded at program start. This can reduce disk activity during the simulation but increases main memory usage. Default: Load only when used.

(V=Vector, F=Float, S=String)

Current restrictions:

- Generic meshes cannot cast shadows. This will be fixed in the future, but shadows will be required to have a convex hull.
- Only one texture is allowed for an object. In the future individual mesh groups will be able to override the global setting.

16.9. Scenario files

Scenarios contain all parameters required to set up the simulation at a particular time. They are used for loading and saving simulation states. Scenario files are usually generated automatically when saving a simulation. The format description below is primarily intended for developers of scenario editor add-ons.

Format:

```
<Description block>
<Environment block>
<Focus block>
<Camera block>
<HUD block>
<Left MFD block>
<Right MFD block>
<Station list>
<Ship list>
```

Description block (optional):

Contains a short description of the scenario.

```
BEGIN_DESC
  <Description>
END_DESC
```

<Description>: ASCII text describing the scenario. CR is converted to space. Empty lines are converted to CR.

Environment block (optional):

Contains the simulation environment.

```
BEGIN_ENVIRONMENT
  <Environment parameters>
END_ENVIRONMENT
```

<Environment parameters>:

Parameter	Type	Description
SYSTEM	String	Name of the planetary system. A configuration file for this system must exist. Default: “Sol”
DATE		Contains simulation start time. Allowed formats are: `MJD` *<mjd>* (*<mjd>*: Modified Julian Date) `JD` *<jd>* (*<jd>*: Julian Date) `JE` *<je>* (*<je>*: Julian Epoch) Default is current simulation time, but this should be avoided if the scenario contains objects defined by position/velocity vectors, which cannot easily be propagated in time.

Focus block (mandatory):

Contains parameters for the user-controlled spacecraft.

```
BEGIN_FOCUS
  <Focus parameters>
END_FOCUS
```

<Focus parameters>:

Parameter	Type	Description
SHIP	String	Name of the user-controlled ship. The ship must be listed in

		the ship list (see below).

Camera block (optional):
Camera mode and parameters. If the camera block is missing, the camera is set to cockpit view in the current focus object.

```
BEGIN_CAMERA
   <Camera parameters>
END_CAMERA
```

<Camera parameters>:

Parameter	Type	Description
MODE	Flag	`Extern` or `Cockpit`
TARGET	String	Camera view target. (external modes only; cockpit mode always refers to current focus object)
POS	Vec_3	Camera position relative to target (external modes only)
TRACKMODE	Flag [+String]	`TargetRelative` | `AbsoluteDirection` | `GlobalFrame` | `TargetTo` *<ref>* | `TargetFrom` *<ref>* (external modes only)
FOV	Float	Field of view (degrees)

HUD block (optional):
HUD mode and parameters. If the HUD block is missing, no HUD is displayed at startup.

```
BEGIN_HUD
   <HUD parameters>
END_HUD
```

<HUD parameters>:

Parameter	Type	Description
TYPE	Flag	`Orbit` | `Surface` | `Docking`

Left/Right MFD blocks (optional):
Left/right MFD type and parameters. If the block is missing, the corresponding MFD is not displayed.

```
BEGIN_MFD Left/Right
   <MFD parameters>
END_MFD
```

<MFD parameters>:

Parameter	Type	Description
TYPE	Flag	MFD type: `Orbit|Surface|Map|Launch|Docking|OAlign|OSync|Transfer`
REF	String	Reference object (Orbit and Map MFD only)
TARGET	String	Target object (for Orbit, OAlign and OSync MFD only)
BTARGET	String	Base target (for Map MFD only)
OTARGET	String	Orbit target (for Map MFD only)
PROJ	Flag	`Ecliptic|Ship|Target` (for Orbit MFD only)
MODE	Flag	`Intersect 1|Intersect 2|Sh periapsis|Sh apoapsis|Tg periapsis|Tg apoapsis|Manual axis` (for OSync MFD only)
MANUALREF	Float	Reference axis position [deg] (for OSync MFD in manual mode only)
LISTLEN	Int	Number or orbit time listings (for OSync MFD only)

Station list:
List of orbital stations. The list is optional, but all stations referred to by the ship list (see below) must exist.

```
BEGIN_STATIONS
```

```
    <Station 0>
    <Station 1>
    . . .
    <Station n-1>
END_STATIONS
```

Station entries: <*Station i*>:

```
<Station name>[:<Class name>]
    <Station parameters>
END
```

<*Station name*>: station identifier string
<*Class name*>: station class (if applicable). If no class is specified, a .cfg file for the station, <*station name*>.cfg is required.
<*Station parameters*>:

Parameter	Type	Description
REF	String	Reference planet/moon the station is orbiting around
EL_MJD	Float	Time in MJD format the orbital elements refer to
ELEMENTS	List	Orbital elements (ecliptic reference) consisting of 6 values: semi-major axis *a* [m], eccentricity *e*, inclination *i* [°], longitude of ascending node Ω [°], longitude of periapsis ϖ [°], mean longitude at EL_MJD *L* [°].
ZROT_FREQ	Float	Rotation frequency for spinning stations. If not specified, the class value is used.
AROT	Vec_3	Orientation: rotation angles of object frame [°]

Ship list:
List of spacecraft. The list must at least contain the vessel referred to by the `Focus` entry.

```
BEGIN_SHIPS
    <Ship 0>
    <Ship 1>
    . . .
    <Ship n-1>
END_SHIPS
```

Ship entries <*Ship i*>:

```
<Vessel name>[:<Class name>]
    <Vessel parameters>
END
```

<*Vessel name*>: ship identifier string
<*Class name*>: vessel class (if applicable). If no class is specified, a .cfg file for the vessel, <*vessel name*>.cfg is required.
<*Vessel parameters*>:

Parameter	Type	Description
STATUS	Flag	`Landed` <*planet*> | `Orbiting` <*planet*> | `Docked` <*planet*>:<*station*>:<*dock*>
BASE		<*base*>:<*lpad*> (only for STATUS Landed)
HEADING	Float	Orientation (only for STATUS Landed)
RPOS	Vec_3	Position rel. to reference (only for STATUS Orbiting)
RVEL	Vec_3	Velocity rel. to reference (only for STATUS Orbiting)
AROT	Vec_3	Orientation: rotation angles of object frame (only for STATUS Orbiting)
VROT	Vec_3	angular velocity [° /s] (only for STATUS Orbiting)
FUEL	Float	Fuel level (0 to 1)
ENGINE_MAIN	Float	Main/retro engine status (-1 to 1)
ENGINE_HOVR	Float	Hover engine status (0 to 1)

17. Planet Surface Textures

17.1. Texture Format

Each planet has an associated texture file <*pname*>.tex, where <*pname*> is the planet's name. The texture file contains a series of texture maps, stored as DirectDraw surfaces (dds) in DXT1 compression format.
ORBITER uses a variable resolution approach for both meshes and texture maps to render planetary surfaces. The rendering resolution level is determined by the apparent radius of the planet. At low resolutions ORBITER uses a single spherical mesh covered by a single texture. At higher resolutions the spherical surface is constructed from a series of sphere patches, each containing its own texture patch. This method allows efficient rendering by removing hidden patches before invoking the rendering pipeline.
ORBITER currently supports 8 resolution levels for planetary surfaces, as listed in Table 1. At the highest resolution the sphere is constructed from 364 patches with an effective texture resolution of 8192x4096. Figure 30 shows a detail of the Martian surface rendered at different resolution levels.

Level	Resolution*	Mesh patches	Triangles (total)**	Texture memory***	
				with DXT1	w/o DXT1
1	64 x 64	1	144	2K	16K
2	128 x 128	1	256	10K	80K
3	256 x 256	1	576	42K	336K
4	512 x 256	2	1024	106K	848K
5	1024 x 512	8	2592	362K	2.9M
6	2048 x 1024	24	4672	1.1M	9.0M
7	4096 x 2048	100	25440	4.3M	34.6M
8	8192 x 4096	364	116448	16.0M	127.8M

Table 1: Supported resolution levels for planetary surfaces.

*Resolution: Effective texture map resolution at the equator.
**Triangles: This is the total number of triangles for all patches. In practice fewer triangles will be rendered because hidden patches are removed before entering the rendering pipeline.
***Texture memory: Video/AGP memory required to hold texture maps up to this resolution level for a single planet. With DXT1: video hardware supports DXT1 texture compression. W/o DXT1: video hardware doesn't support DXT1 texture compression.

High resolution levels require significant amounts of video/AGP memory and should only be used on systems with adequate 3D graphics subsystems. On older graphics cards which do not natively support DXT1 texture compression ORBITER needs to convert textures into RGBA format which increases memory requirements 8-fold. Conversion to RGBA will also dramatically increase the loading time when starting ORBITER.

Important: Do not try to use resolution level 8 if your video card does not support DXT1 texture compression or has less than 32MB of texture memory!

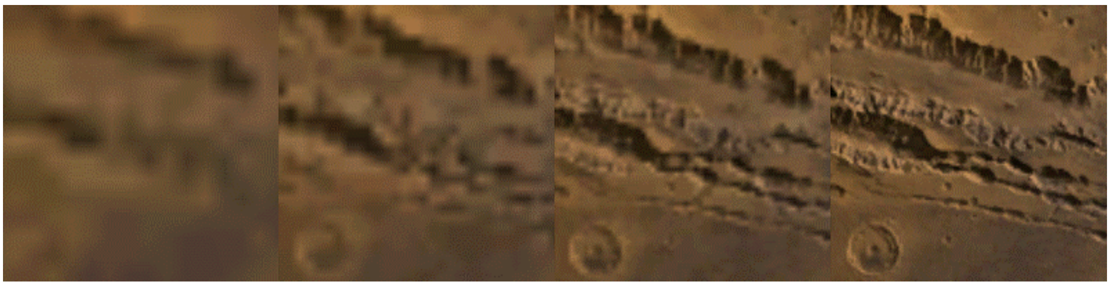


Figure 30: Mars texture detail at resolution levels 5, 6, 7 and 8 (from left).

17.2. Where ORBITER looks for textures

ORBITER first searches for the texture file in the location specified by the `HightexDir` entry in the *Orbiter.cfg* file (see section 16.1). If the texture file is not found or if `HightexDir` is not defined then ORBITER searches in the directory specified by the `TextureDir` entry. This allows switching between high and low resolution texture maps conveniently by inserting or removing the `HightexDir` entry.

If no texture file is found then the planet is rendered without a surface texture.

Each planet's configuration file <*pname*>.cfg contains an entry `MaxPatchResolution` which defines the maximum texture resolution level to use with this planet (valid range 1 to 8). If the texture file contains higher resolution levels than defined by `MaxPatchResolution` then the additional resolutions are skipped. This allows reducing texture memory requirements without modifying the texture file. If the texture file contains fewer resolution levels than defined by `MaxPatchResolution` then the maximum resolution is reduced accordingly.

17.3. Using *pltex* to Generate Custom Planet Textures

If you prefer, you can use your own planet maps instead of those provided by ORBITER. The ORBITER download page contains a planet texture conversion tool (*pltex*) which allows to convert planet maps from BMP bitmap format to ORBITER's texture format. It resamples the map to the requested resolutions, splits it into surface patches and converts them to DXT1 compressed texture format.

The source map should contain the complete surface in spherical projection, where the left edge corresponds to longitude 180° W, the right edge to longitude 180° E, the bottom edge to latitude 90° S, and the top edge to latitude 90° N. The width/height ratio of the bitmap should be close to 2/1.

Pltex requires the source map in 24bit or 8bit Windows BMP format. If your map is in any other format (e.g. JPEG or GIF) you need to convert it into BMP (using your favourite graphics conversion tool) before invoking pltex.

Synopsis:

`pltex [–i` <*mapname*>`] [-l` <*minres*> `-h` <*maxres*>`]`

<*mapname*>*:* source texture file name
<*minres*>*:* minimum resolution level (1..8)
<*maxres*>*:* maximum resolution level (<*minres*>..8)

- If command line options are omitted then pltex requests values interactively.
- If a higher maximum resolution is requested than can be obtained from the source map, pltex adjusts the maximum resolution accordingly. See Table 1 for map resolutions at the various resolution levels.
- The only justification for <*minres*> ≠ 1 is if you want to compose certain resolution levels from a different source map, e.g. generate Earth resolution levels 1 to 7 from a map that includes clouds, and level 8 from a map without clouds. In that case pltex must be run twice, and the output texture files concatenated.
- The option to use alpha (transparency) maps is intended for semi-transparent cloud maps.

Pltex will generate a texture file <*mapname*>.tex. If necessary, rename to <*pname*>.tex where <*pname*> is the planet's name, and copy to the `TextureDir` or `HightexDir` directory (see section 16.1).

Note:
To generate resolution level 8 from a 8192x4096 planet map you need at least 128MB of system memory.

PS: If you find a good quality high-resolution public domain planet map on the web, let me know, and I'll put it into the ORBITER distribution.

Appendix A. Mesh file format

This section has been moved to 3DModel.doc in the Orbitersdk\doc subdirectory, which is contained in the SDK package.

Appendix B. Solar System: Constants and Parameters

Astrodynamic Constants and Parameters

Constant	Symbol	Value
Julian day	d	86400 s
Julian year	yr	365.25 d
Julian century	Cy	36525 d
Speed of light	c	299792458 m/s
Gaussian gravitational constant	k	0.01720209895 $(AU^3/d^2)^{1/2}$

Table 2: Defining constants

Constant	Symbol	Value
Mean siderial day		86164.09054 s = 23:56:04.09054
Sidereal year (quasar ref. frame)		365.25636 d
Light time for 1 AU	τ_A	499.004783806 (± 0.00000001) s
Gravitational constant	G	6.67259 (± 0.00030) × 10^{-11} kg^{-1} m^3 s^{-2}
General precession in longitude		5028.83 (± 0.04) arcsec/Cy
Obliquity of ecliptic (J2000)	ε	84381.412 (± 0.005) arcsec
Mass: Sun / Mercury		6023600. (± 250.)
Mass: Sun / Venus		408523.71 (± 0.06)
Mass: Sun / (Earth+Moon)		328900.56 (± 0.02)
Mass: Sun / (Mars system)		3098708. (± 9.)
Mass: Sun / (Jupiter system)		1047.3486 (± 0.0008)
Mass: Sun / (Saturn system)		3497.898 (± 0.018)
Mass: Sun / (Uranus system)		22902.98 (± 0.03)
Mass: Sun / (Neptune system)		19412.24 (± 0.04)
Mass: Sun / (Pluto system)		1.35 (± 0.07) × 10^8
Mass: Moon / Earth		0.012300034 (± 3 × 10^{-9})

Table 3: Primary constants

Constant	Symbol	Value
Astronomical unit distance	$c \times \tau_A$ = AU	1.49597870691 × 10^{11} (± 3) m
Heliocentric gravitational constant	$k^2\ AU^3\ d^{-2} = GM_{sun}$	1.32712440018 × 10^{20} (± 8 × 10^9) m^3 s^{-2}
Mass: Earth / Moon		81.30059 (± 0.00001)

Table 4: Derived constants

Notes:
Data are from the 1994 IAU file of current best estimates. Planetary ranging determines the Earth/Moon mass ratio. The value for 1 AU is taken from JPL's current planetary ephemeris DE-405.

Reference:
Standish, E.M. (1995) ``Report of the IAU WGAS Sub-Group on Numerical Standards'', in Highlights of Astronomy (I. Appenzeller, ed.), Table 1, Kluwer Academic Publishers, Dordrecht.

Planetary Mean Orbits (J2000)

(Epoch = J2000 = 2000 January 1.5)

Planet (mean)	a [AU]	e	i [deg]	Ω [deg]	ϖ [deg]	L [deg]
Mercury	0.38709893	0.20563069	7.00487	48.33167	77.45645	252.25084

Venus	0.72333199	0.00677323	3.39471	76.68069	131.53298	181.97973
Earth	1.00000011	0.01671022	0.00005	-11.26064	102.94719	100.46435
Mars	1.52366231	0.09341233	1.85061	49.57854	336.04084	355.45332
Jupiter	5.20336301	0.04839266	1.30530	100.55615	14.75385	34.40438
Saturn	9.53707032	0.05415060	2.48446	113.71504	92.43194	49.94432
Uranus	19.19126393	0.04716771	0.76986	74.22988	170.96424	313.23218
Neptune	30.06896348	0.00858587	1.76917	131.72169	44.97135	304.88003
Pluto	39.48168677	0.24880766	17.14175	110.30347	224.06676	238.92881

Table 5: Planetary mean orbits

Planetary Orbital Element Centennial Rates

(for the mean elements given above)

Planet (rate)	a [AU/Cy]	e [1/Cy]	i [“/Cy]	Ω [“/Cy]	ϖ [“/Cy]	L [“/Cy]
Mercury	0.00000066	0.00002527	-23.51	-446.30	573.57	538101628.29
Venus	0.00000092	-0.00004938	-2.86	-996.89	-108.80	210664136.06
Earth	-0.00000005	-0.00003804	-46.94	-18228.25	1198.28	129597740.63
Mars	-0.00007221	0.00011902	-25.47	-1020.19	1560.78	68905103.78
Jupiter	0.00060737	-0.00012880	-4.15	1217.17	839.93	10925078.35
Saturn	-0.00301530	-0.00036762	6.11	-1591.05	-1948.89	4401052.95
Uranus	0.00152025	-0.00019150	-2.09	-1681.40	1312.56	1542547.79
Neptune	-0.00125196	0.0000251	-3.64	-151.25	-844.43	786449.21
Pluto	-0.00076912	0.00006465	11.07	-37.33	-132.25	522747.90

“	arcsecs
Cy	Julian century
a	Semi-major axis
e	eccentricity
i	inclination
Ω	longitude of the ascending node
ϖ	longitude of perihelion
L	mean longitude

Notes:
This table contains mean orbit solutions from a 250 yr. least squares fit of the DE 200 planetary ephemeris to a Keplerian orbit where each element is allowed to vary linearly with time. This solution fits the terrestrial planet orbits to ~25" or better, but achieves only ~600" for Saturn. Elements are referenced to mean ecliptic and equinox of J2000 at the J2000 epoch (2451545.0 JD).

Reference:
Explanatory Supplement to the Astronomical Almanac. 1992. K. P. Seidelmann, Ed., p.316 (Table 5.8.1), University Science Books, Mill Valley, California.

Planets: Selected Physical Parameters

Planet	Mean radius [km]	Mass [10^{23}kg]	Density [g/cm^3]	Siderial rotation period [h]	Siderial orbit period [yr]
Mercury	2440. ±1.	3.302	5.427	1407.509	0.2408445
Venus	6051.84 ±0.01	48.685	5.204	-5832.444	0.6151826
Earth	6371.01 ±0.02	59.736	5.515	23.93419**	0.9999786
Mars	3389.92 ±0.04	6.4185	3.9335±0.0004	24.622962	1.88071105
Jupiter	69911. ±6.	18986.	1.326	9.92425	11.856523
Saturn	58232. ±6.	5684.6	0.6873	10.65622	29.423519
Uranus	25362. ±12.	868.32	1.318	17.24 ±0.01	83.747407
Neptune	24624. ±21.	1024.3	1.638	16.11 ±0.01	163.72321
Pluto	1151*	0.15*	1.1*	153.28*	248.0208*

Planet	V(1,0) [mag.]	Geometric albedo	Equatorial gravity [m/s^2]	Escape velocity [km/s]
Mercury	-0.42	0.106	3.701	4.435
Venus	-4.4	0.65	8.87	10.361
Earth	-3.86	0.367	9.780327	11.186
Mars	-1.52	0.15	3.69	5.027

Jupiter	-9.4	0.52	23.12 ± 0.01	59.5
Saturn	-8.88	0.47	8.96 ± 0.01	35.5
Uranus	-7.19	0.51	8.69 ± 0.01	21.3
Neptune	-6.87	0.41	11.00 ± 0.05	23.5
Pluto	-1.0*	0.3*	0.655	1.3

(all values from reference 1. except (*) from reference 2)
** Orbiter now uses 23.93447h (= 23h 56m 4.09s) which appears to give better long term stability.

References
1. Yoder, C.F. 1995. ``Astrometric and Geodetic Properties of Earth and the Solar System'' in Global Earth Physics, A Handbook of Physical Constants, AGU Reference Shelf 1, American Geophysical Union, Tables 6,7,10.

2. Explanatory Supplement to the Astronomical Almanac. 1992. K. P. Seidelmann, Ed., p.706 (Table 15.8), University Science Books, Mill Valley, California.

Rotation elements

Planet	**North pole**		**Obliquity of ecliptic [°] (*)**	**Longitude of Sun's transit [°] (*)**
	Right ascension α_1 [°]	**Declination δ_1 [°]**		
Mercury	280.99	61.44	7.01	228.31
Venus	272.78	67.21	1.27	302.07
Earth	-	90	23.44	0
Mars	317.61	52.85	26.72	262.78
Jupiter	268.04	64.49	2.22	157.68
Saturn	40.14	83.50	28.05	349.39
Uranus	257.29	-15.09	82.19	167.62
Neptune	295.25	40.63	29.48	221.13
Pluto	311.50	4.14	68.69	225.19

Reference:
The Astronomical Almanac 1990 (North pole coordinates)

(*) Derived from north pole coordinates (MS)

Atmospheric Parameters

Planet	**Surface pressure [kPa]**	**Surface density [kg/m^3]**	**Scale height [km]**	**Avg. temp [K]**	**Wind speeds [m/s]**
Mercury					
Venus	9200	~65	15.9	737	0.3-1 (surface)
Earth	101.4	1.217	8.5	288	0-100
Mars	0.61 (variable)	~0.020	11.1	~210	0-30
Jupiter	>> 10^4	~0.16 at 1 bar	27	~129 ~165 at 1 bar	up to 150 at < 30° latitude up to 40 else
Saturn	>> 10^4	~0.19 at 1 bar	59.5	~97 ~134 at 1 bar	up to 400 at < 30° latitude up to 150 else
Uranus	>> 10^4	~0.42 at 1 bar	27.7	~58 ~76 at 1 bar	0-200
Neptune	>> 10^4	~0.45 at 1 bar	19.1-20.3	~58 ~72 at 1 bar	0-200
Pluto					

Appendix C. Calculation of Orbital Elements

Six scalar parameters ("elements") are required to define the shape of an elliptic orbit, its location in space and a location along its trajectory.

a Semi-major axis
e Eccentricity

i Inclination
Ω Longitude of ascending node
ω argument of periapsis
υ true anomaly

Calculating Elements from Position and Velocity Vectors

Let $\mathbf{r}$ and $\mathbf{v}$ be the cartesian position and velocity vectors of an orbiting object in coordinates of a reference frame with respect to which the elements of the orbit are to be calculated (e.g. geocentric equatorial for an orbit around Earth, or heliocentric ecliptic for an orbit around the Sun). We assume a right-handed system with the x-axis pointing towards the vernal equinox (or other reference direction) and the z-axis pointing upwards.
Compute the following auxiliary vectors:

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} = (r_y v_z - r_z v_y, -r_x v_z + r_z v_x, r_x v_y - r_y v_x)$$

$$\mathbf{n} = \mathbf{z} \times \mathbf{h} = (-h_y, h_x, 0)$$

$$\mathbf{e} = \frac{1}{\mu}\left[\left(v^2 - \frac{\mu}{|\mathbf{r}|}\right)\mathbf{r} - (\mathbf{r} \cdot \mathbf{v})\mathbf{v}\right]$$

where $\mathbf{h}$ is a vector perpendicular to the orbital plane, $\mathbf{n}$ points towards the ascending node (the z-component of $\mathbf{n}$ is zero), and $\mathbf{e}$ is the eccentricity vector with $\mu = GM$ (pointing towards the periapsis), G is gravitational constant and M is the mass of the central body (neglecting the mass of the orbiter).

Semi-major axis:

$$a = \frac{-\mu}{2E} \quad \text{with} \quad E = \frac{v^2}{2} - \frac{\mu}{|\mathbf{r}|}$$

Eccentricity:

$$e = |\mathbf{e}| \quad \text{or} \quad e = \sqrt{1 + \frac{2Eh^2}{\mu^2}}$$

Inclination:

$$i = \arccos\frac{h_z}{|\mathbf{h}|}$$

Longitude of ascending node:

$$\Omega = \arccos\frac{n_x}{|\mathbf{n}|} \quad (\text{if } n_y < 0 \text{ then } \Omega = 2\pi - \Omega)$$

Ω is the angle between reference direction (1,0,0) (e.g. vernal equinox) and the ascending node.
Ω is undefined for equatorial orbits (i = 0), in which case ORBITER by convention sets Ω = 0, i.e. it places the ascending node in the reference direction, which is equivalent to setting $\mathbf{n}/|\mathbf{n}| = (1,0,0)$.

Argument of periapsis:

$$\omega = \arccos\frac{\mathbf{n} \cdot \mathbf{e}}{|\mathbf{n}||\mathbf{e}|} \quad (\text{if } e_z < 0 \text{ then } \omega = 2\pi - \omega)$$

ω is the angle between the ascending node and the periapsis. ω is undefined for equatorial orbits in which case according to above convention we get

$$\omega = \arccos\frac{e_x}{|\mathbf{e}|} \quad (\text{if } e_z < 0 \text{ then } \omega = 2\pi - \omega)$$

ω is also undefined for circular orbits in which case ORBITER by convention places the periapsis at the ascending node, i.e. $\omega = 0$.

True anomaly:

$$\upsilon = \arccos \frac{\mathbf{e} \cdot \mathbf{r}}{|\mathbf{e}||\mathbf{r}|} \text{ (if } \mathbf{r} \cdot \mathbf{v} < 0 \text{ then } \upsilon = 2\pi - \upsilon\text{)}$$

υ is the angle between the periapsis and object position. Note that this expression is undefined for circular orbits, in which case the periapsis coincides with the ascending node according to the convention above, i.e.

$$\upsilon = \arccos \frac{\mathbf{n} \cdot \mathbf{r}}{|\mathbf{n}||\mathbf{r}|} \text{ (if } \mathbf{n} \cdot \mathbf{v} > 0 \text{ then } \upsilon = 2\pi - \upsilon\text{)}$$

If in addition the inclination is zero then the true anomaly further simplifies to

$$\upsilon = \arccos \frac{r_x}{|\mathbf{r}|} \text{ (if } v_x > 0 \text{ then } \upsilon = 2\pi - \upsilon\text{)}$$

Some dependent parameters can be derived from the above elements:

Linear eccentricity:

$\varepsilon = a\,e$

Semi-minor axis:

$b^2 = a^2(1 - e^2)$

Periapsis and apoapsis distances:

$d_p = a(1 - e)$

$d_a = a(1 + e)$

Longitude of the periapsis:

$\varpi = \Omega + \omega$

Eccentric anomaly:

$$E = \arccos \frac{1 - |\mathbf{r}|/a}{e}$$

Mean anomaly:

$M = E - e \sin E$

Mean longitude:

$L = M + \varpi$

True longitude:

$l = \varpi + \upsilon$

Orbit period:

$T = 2\pi\sqrt{a^3/\mu}$